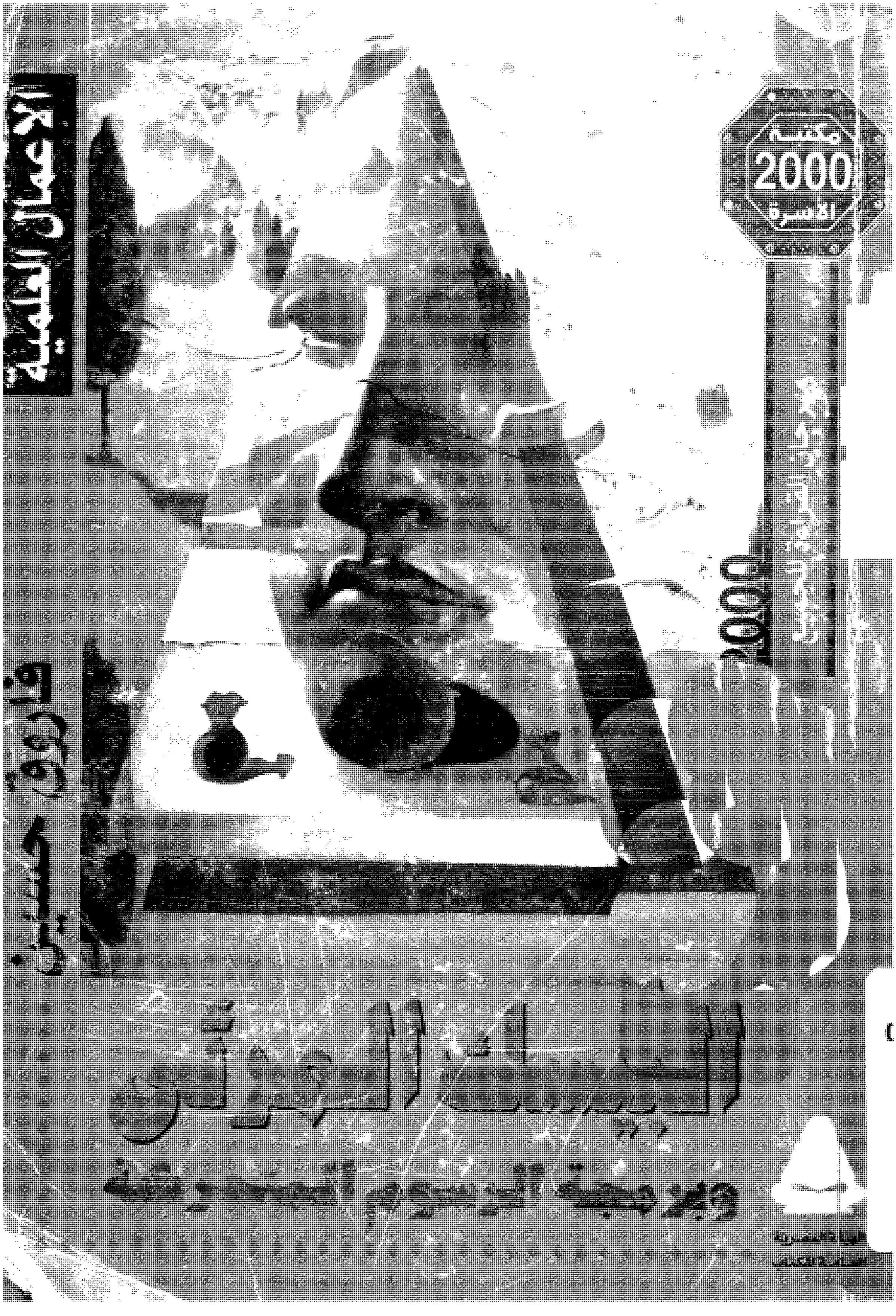

الأعمال العلمية
مكتبة الأسرة
2000
مهرجان القراءة للجميع
2000
فاروق حسين
البيسك العربي
وبرمجة الرسوم المتحركة
الهيئة المصرية العامة للكتاب

البيسيك المرئى وبرمجة
الرسوم المتحركة

اسم العمل: أفروديت فى أرض الهروب ١٩٨١م

سلفادور دالى (١٩٠٤ - ١٩٨٩)

مصور أسبانى، انشغل فى بداية أمره بممارسة الفن التجريدى، وانتقل من أسبانيا إلى باريس ليستقر بها منذ عام ١٩٢٩م، ثم رحل إلى أمريكا منذ عام ١٩٤٠م، وقد قدم للسريالية كشوفات جديدة، وهو الذى زاول واستخدم الأسلوب التكعيبى والمستقبلى، وقد صور لوحات بديعة عن الأحلام وما يجول ويصول فى العقل الباطن، كما صور فى لوحاته اللامعقول والانفعالات. ومن أهم ما يميزه، تلك الجرأة التى أضافها للفن السريالى فى أفلام السينما والاستعراضات الراقصة والإعلانات.

محمود الهندى

البيسيك المرئى وبرمجة الرسوم المتحركة

المهندس/ فاروق سيد حسين

مهرجان القراءة للجميع ٢٠٠٠

مكتبة الأسرة

برعاية السيدة سوزان مبارك

(الأعمال العلمية)

البيسيك المرئى وبرمجة
الرسوم المتحركة

المهندس/ فاروق سيد حسين

الغلاف
والإشراف الفنى:
الفنان : محمود الهندى

المشرف العام :
د . سمير سرحان

الجهات المشاركة:
جمعية الرعاية المتكاملة المركزية
وزارة الثقافة
وزارة الإعلام
وزارة التعليم
وزارة الإدارة المحلية
المجلس الأعلى للشباب والرياضة
التنفيذ : هيئة الكتاب

على سبيل التقديم

«كتاب لكل مواطن ومكتبة لكل أسرة» تلك الصيحة التى أطلقتها المواطنة المصرية النبيلة «سوزان مبارك» فى مشروعها الرائع «مهرجان القراءة للجميع ومكتبة الأسرة» والذى فجر ينابيع الرغبة الجارفة للثقافة والمعرفة لشعب مصر الذى كانت الثقافة والابداع محور حياته منذ فجر التاريخ.

وفى مناسبة مرور عشر سنوات على انطلاق المشروع الثقافى الكبير وسبع سنوات من بدء مكتبة الأسرة التى أصدرت فى سنواتها الست السابقة «١٧٠٠» عنواناً فى حوالى «٣٠» مليون نسخة لاقت نجاحاً واقبالاً جماهيرياً منقطع النظير بمعدلات وصلت إلى «٣٠٠» ألف نسخة من بعض إصداراتها.

وتنطلق مكتبة الأسرة هذا العام إلى آفاق الموسوعات الكبرى فتبدأ بإصدار موسوعة «مصر القديمة» للعلامة الاثرى الكبير «سليم حسن» فى «١٦» جزءاً إلى جانب السلاسل الراسخة «الابداعية والفكرية والعلمية والروائع وامهات الكتب والدينية والشباب» لتحاول أن تحقق ذلك الحلم النبيل الذى تقوده السيدة: سوزان مبارك نحو مصر الأعظم والأجمل.

د. سمير سرحان

☆ المقـــدمة ☆

لغة البيسيك المرئية هى أكثر اللغات الشائعة لعمل برامج WinCGI . وهى لغة مؤسسة على أهداف (objects) والتى تدعم كثيراً من المبادىء ، مثل البرامج الفعلية الموجهة لهدف (object - Oriented Programming : oop) وهذا يعنى أن برامجنا تحتوى على أشياء ، مثل التحكمات والأشكال وبرامج أخرى ، وأشياء هدف مثل الطابعات وهياكل الملفات.

كذلك يوجد مشترك مستخدم الرسومات (: Graphics User Interface GUI) والذى يساعد كثيراً فى أداء الرسومات باختصار كثير من البرامج. ونرى فى هذا الكتاب كيفية استخدامها فى عمل الرسوم المتحركة .

وقد شرحت معانى بعض التعبيرات الانجليزية والتى قد يستعصى على القارىء فهمها بعد هذا التعبير مباشرة بقدر المستطاع حتى أتيح للقارىء الاستفادة القصوى ويسهل الاستعانة بالمراجع الأجنبية .

وتم ترجمة بعض التعبيرات الانجليزية بعد انتهاء الجملة لصعوبة ترجمتها فى الحال لاحتوائها على كثير من الكلمات . وأرجو أن أكون قد وفقت فى هذا العرض ولك منى كل تحية وتقدير عزيزى القارىء.

م / فاروق سيد حسين

الباب الأول

لغة البيسيك المرئية

الباب الأول
لغة البيسيك المرئية

اللغة الشائعة جداً لإنشاء برامج WinCGI هى البيسيك المرئية ، وهى لغة مؤسسة على شىء (object) والتى تدعم كثيراً من الأسس المماثلة للغات OOP الفعلية (برمجة موجهة لشىء object - oriented Programming) . وهذا يعنى أن برامجنا تحتوى على أهداف مثل التحكمات (Controls) ، والأشكال (Forms) وبرامج أخرى وأشياء نظم أخرى مثل الطابعات والهياكل . البيسيك المرئية تجمع الأشياء ، وتجعل تصميم البرنامج أسرع وأكثر كفاءة ، وتمكننا من إعادة استعمال الأشياء المنشأة فى برامجنا الأخرى أو الوصول لأشياء متوفرة لدينا (Off - shelf) كثيرة، (multitude) متوفرة فعلاً .

ملاحظة : كلا ميكروسوفت وبائعين طرف ثالث تبيع وتوزع قطع مجانية للشفرة بامتداد VBX. أو OCX. هذه الأشياء تؤدى وظائف محددة يمكن تضمينها فى برنامجنا ، لتوفير ساعات التشغيل .

وقد أصبحت بيسيك المرئية لغة WinCGI محببه نتيجة انتشارها كحل برمجة مشتركة (Corporate) . وإذا كان لدينا مبرمجين لبيسيك المرئية فى قسمنا IS ؛ فيمكن أن ترد بسرعة للإسراع فى إنشاء تطبيقات شبكة (Web) طرف خلفى (back - end) تطبيق إدارة قاعدة بيانات.

*** تطبيق إدارة قاعدة بيانات : back - end**

قراءة وعرض HTML :

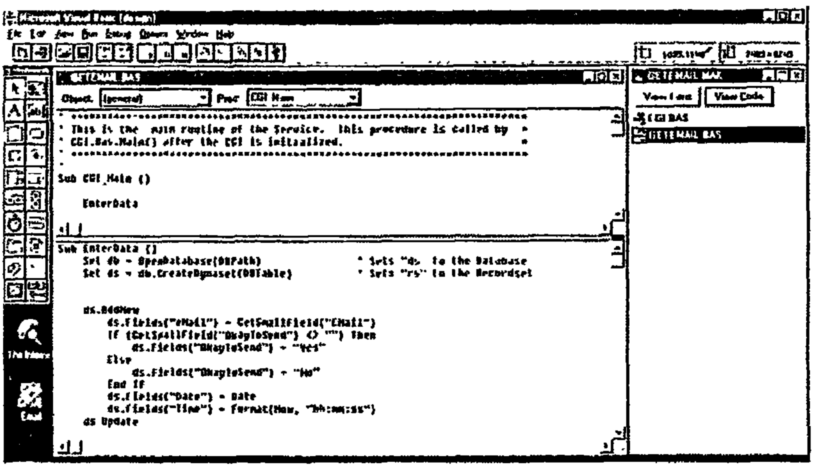
العمل الأكثر وضوحاً والـذى نحتـاج أن نؤديـه عندمـا نتعامل مع خادم شبكة (Web server) ومتصفح عميل هو قراءة عرض شفرة HTML . ولمساعدتنا فى إتمام ذلك، يوجد ملف يسمى cgi 32. bas يمكن الحصول عليه من مواقع متعددة على الانترنيت، يتضمن Inc ، Associates & O' Reilly. الملف يتضمن وظائف لقراءة وعرض صفحات HTML. هذه الوظائف يمكن أن تساعدنا على بناء مشترك HTML . ونحن مقيدون إلى حد ما بمشترك HTML لأن ربطنا بخادم الشبكة ليس أبّهة (Stateless) ، وهذا يعنى أننا فى كل مرة نقرأ أو نرسل معلومة للخادم (Server) ، يتم الربط ، وتمر المعلومة ، والربط يقفل . ولا يمكننا الاعتماد على البيانات التى أرسلناها أو استقبلناها على ربطنا الأخير بالخادم ، لأن ذلك الربط قد تم قفله . ولم يعد يحتفظ (retains) بأى مراجع مع الربط السابق . وكل ربط مع الخادم يجب أن يكون عملاً بينياً محتوى / ذاتى. وإذا أردنا حفظ محادثة متطورة (ongoing) مع الخادم ، يجب أن نحفظ نوعاً ما لملف مؤقت (temp) لاختزان البيانات .

الطريقة الأخرى لحفظ محادثه مع الخادم هى طمر (embed) بيانات على صفحتنا كحقول مختفية أو بعد أن نحصل على معلومات من الخادم أو مستخدم قد يكون ضرورياً لمعالجة ما، والتى قد تأخذ مكان صفحات متعددة . من الآن ، يمكننا إختزان البيانات فى حقول مختفية . ونمرر البيانات للخلف وللأمام بين الخادم والمتصفح خلال الحقول المختفية حتى نصل للصفحة التى تحتاج تلك المعلومة عند النقطة التى قرأناها من الحقول المختفية.

تنفيذ بيسيك المرئية :

نحتاج للحصول على نسخة للغة برمجة بيسيك المرئية لميكروسوفت . هذه الباقة

تتضمن بيئة تطوير متكاملة والمبينة فى شكل (١ ـ ١) ، ليشمل معظم الأدوات المطلوبة لانشاء تطبيق (back - end) . بالاضافة لذلك ، نحتاج الملف لدعم تطبيق المسمى WinCGI (أو cgi 32. bas لاصدارات 32 رقم ثنائى لبيسيك المرئية) . يمكننا الحصول على هذا الملف من عدد من المواقع على الانترنيت ، متضمنة(http // : software . orma. com Inc, Associates, O'Reilly) فهو يحتوى على كل الوظائف المطلوبة للعمل البينى فى بيئة HTML.



بيئة تطوير بيسيك مرئية لميكروسوفت

برنامج عينة :

المثال الموجود فى الأجزاء التالية ، مكتوب ببيسيك المرئية ، يبين لنا كيف يمكننا استخراج بيانات من الشكل الذى أنشأناه . يمكن وضع هذا الشكل على موقعنا فى الشبكة للزائرين لملئه. حينئذ ، فإن "WinGGI "back end يأخذ معلومة عن الناس الذين يدخلون على الشكل ويحفظه فى ملف أو يرسله بالبريد الالكترونى (e - mails) لشخص محدد .

إنشاء شكل (form)

يمكن استعمال الشكل الذى سبق أن أنشأناه ، أو يمكننا إنشاء واحد جديد قبل أن نبدأ هذا الإجراء . ويمكن وضع، أى نوع من تحكم الشكل على الشكل ، متضمنة صناديق التعديل(edit boxes) ،وصناديق قوائم ، وأزرار راديو وصناديق فحص . خدمة الطرف الخلفى يمكنها قراءة البيانات من كل تحكمات الشكل هذه.

إضافة حقول مختفية :

حتى يمكن لشكلنا أن يعمل مع خدمة الطرف الخلفى (back - end) التى سننشؤها ، نحتاج أولا لنضيف حقول قليلة مختفية لها .فى الحقيقة ، هذه الحقول تعطى طرق جمع البيانات . وهنا ، كيف نضيف الحقول :

١ ـ نفتح شكلنا فى معدل HTML .

٢ ـ نحرك العلامة (Cursor) لموقع فى مكان ما تحت عروة (Tag) الـ < FORM > ونتأكد أننا لسنا فى منتصف أى عراوى .

٣ ـ الحقل الأول الذى نحتاج أن نضيفه يخبر الطرف الخلفى إذا كنا نريد حفظ البيانات التى أدخلها مستخدم فى ملف . ندخل ما يلى إذا كنا نريد حفظ معلومات المستخدم فى ملف :

< INPUTTYPE = HIDDEN NAME = " Save - T - File " VAL-
UE = " Yes "

ثم نذهب للخطوة ٤ .. كذلك ، يمكننا إدخال ما يلى إذا لم نكن نرغب المعلومات المختزنة فى ملف

< INPUT TYPE = HIDDEN NAME = " Save - T - File " VAL-
UE = " NO"

ثم نقفذ للخطوة « ٥ » .

٤ ـ لأننا نريد المعلومات المختزنة فى الملف ، يجب أن نخبر الخدمة أن تنشىء الملف وتكتب له النص . ندخل ما يلى :

< INPUT TYPE = HIDDEN NAME = " Data File " VALUE" =
"C: / MyDir / Form X . CsV" >

ويجب أن نضع مسارنا الخاص وإسم الملف فى جزء VALUE للمعلومة . فمثلاً ، استطعنا إنشاء دليل على خادم الشبكة يسمى " Form Data " حينئذ ، يمكننا نسخ الملف لهذا الدليل . إذا استعملنا الاسمForm Data ، فإن الحقل المختص سيبدو مثل هذا :ـ

< INPUT TYPE = HIDDEN NAME = " Data - File " VALUE =
"C \ Form Data \ Form X . CSV " >

وإذا كان يجب أن نؤدى العمل الذى وصف فى الخطوة السابقة ، نتحرك للخطوة ٧.

٥ ـ إذا أردنا إرسال المعلومة فى شكل form لواحد أو أكثر من الناس مع بريد الكترونى ، فنحتاج لاضافة المدخل التالى :

< INPUT TYPE = HIDDEN NAME = " Mail Information "
VALUE = " Yes "

ثم نذهب للخطوة رقم ٦ . إذا أردنا أن نرسل المعلومة بالبريد الالكترونى (e - mail) ، ندخل ما يلى :

< INPUT TYPE = HIDDEN NAME = " Mail Information "
VALUE = " NO " >

ونذهب للخطوة ٧ .

٦ ـ لأننا قد قررنا أن نرسل المعلومة بالبريد ، فنحتاج لاضافة حقول بحيث يمكن معالجة البريد . هذه الحقول هى (REQUIRED) إذا كنا قد قررنا إرسال المعلومة بالبريد Mail Information.

ملاحظة : إرسال المعلومة بالبريد يعمل مع خادم البريد SMTP فقط.
ندخل ما يلى :

< INPUT TYPE = HIDDEN NAME = " Mail - Server - Name "
VALUE = "Mailserver . Your Company . Com " >

هذا الحقل يجب أن يحتوى على الاسم أو عنوان IP لخادم البريد الخاص بنا فى جزء VALUE. فمثلاً ، يمكن أن ندخل :

يلى ذلك ، ندخل

" **ns . frontiertech . com** " or **192 . 104 . 32 . 3** . Next, enter

< INPUT TYPE = HIDDEN NAME = " Mail - To - Address "
VALUE = " Joesmitha Your Company . Com " >

جزء VALUE لهذا الحقل يجب أن يحتوى على العنوان الذى نريد إرسال المعلومة له. (إذا أردنا ذلك ، بدلاً من ذلك يمكننا جعل هذا الحقل جزئى . يمكننا إنشاء صندوق قائمة مرئى يحتوى على أسماء الناس أو أسماء المنتجات . وقيمة كل بند حينئذ يمكن أن تكون عنوان بريد الإلكترونى) . الآن ، ندخل ما يلى :

< INPUT TYPE = HIDDEN NAME = " Mail - Server - Login "
VALUE = "Joe Smith " >

هـذا الحقل يجب أن يحتوى عـلى إسـم المستخدم لشخص ما على خادم بريدنا

(من المحتمل The Webmaster أو مدير نظام) . البريد الذى تم إرساله ويشبهه أتى من هذا الشخص .

٧ ـ سواء كتبنا معلومة الشكل للملف أو أرسلناها بالبريد الالكترونى ، يجب أن ندخل ما يلى :

< INPUT TYPE = HIDDEN NAME = " Field Names " VALUE = "First Name - Last Name - Address - City - State - Zip " >

هذا الحقل يخبر الخدمة أى الحقول على شكلنا . وإذا لم ندون حقل هنا ، فهو لم يقرأ أو يعالج . ونحتاج لادخال كل أسماء الحقل مفصولة بـــ فى جزء VALUE. إذا لم نكن متأكدين ما هى أسماء الحقل ، يمكننا إيجادها بالنقل خلفياً إلى الـ HTML. طول Edit (التعديل) لـ Web Designer والطقطقة الصحيحة على حقل . (لا ننسى : إذا استعملنا أزرار راديو مثلاً ، على شكلنا ، نحتاج لادخال إسم المجموعة Group Name للأزرار، وليست قيمة كل زر) .

٨ ـ بقية العراوى (Tags) التى يمكننا إضافتها غير مطلوبة ولكن يمكنها جعل صفحة إستجابتنا أكثر تفرداً. وهنا بعض الأمثلة :

● يمكننا إدخال حقل يخبر الخدمة أى نص (text) يوضع فى قضيب العنوان (title lebar) فى صفحة الاستجابة ، مثل :

< INPUT TYPE = HIDDEN NAME = "Response - Title "VALUE=" Thank You Page">

● كذلك ، يمكننا إدخال الحقل التالى والذى يخبر الخدمة أن النص الذى سيعرض على صفحة الاستجابة :

< INPUT TYPE = HIDDEN NAME = "Response Text" - VALUE =" Thank You FOR FILLING OUT OUR FORM." >

● يمكننا تضمين حقل ، مثل التالى ، والذى يخبر الخدمة أى وصلة توضع على صفحة الاستجابة .

< INPUT TYPE = HIDDEN NAME = "Response - Link" VALUE = "Http: // www. Your Server. com " >.

● أخيراً ، يمكننا إضافة حقل والذى يخبر الخدمة أى نص يدخل لوصلة الاستجابة « Response - Link » ، مثل :

< INPUT TYPE = HIDDEN NAME = "Response- Link - Text" VALUE = "Go back to our Hom page." > .

٩ ـ عندما نكون قد أنهينا إضافة حقول ، نتأكد من حفظ الشكل (Form) لخادم شبكتنا . يلى ذلك ، نحاول الوصول للشكل خلال متصفح لرؤية كيف يعمل (مثلاً)، ندخل . (http : / / yourserver. com / formpage. htm) .

☐ الشكل المستكمل :

هنا كيف يشبه شكل العينة ، إذا ضمننا كل الحقول المطلوبة:

```
<H1>
Please enter your information below
</H1>
<HR SIZE=5>
<FORM METHOD=POST ACTION="/FORMHELP">

<!--        This is the section of hidden fields

<INPUT TYPE=HIDDEN NAME="Save_To_File" VALUE="Yes">
<INPUT TYPE=HIDDEN NAME="Data_File" VALUE="C:\FormData\Sample.csv">
<INPUT TYPE=HIDDEN NAME="Mail_Information" VALUE="No">
<INPUT TYPE=HIDDEN NAME="Response_Title" VALUE="Sample Form">
<INPUT TYPE=HIDDEN NAME="Response_Text" VALUE="Thank you for your use of this sample form.">
<INPUT TYPE=HIDDEN NAME="Response_Link" VALUE="http://www.YourServer.com/Homepage.htm">
<INPUT TYPE=HIDDEN NAME="Response_Link_Text" VALUE="Back to our home page.">
<INPUT TYPE=HIDDEN NAME="Field_Names" VALUE="FirstName~LastName~Address~City~State~Zip~Phone~E-mail">
<!--        End of the hidden fields        -->

<PRE>
First Name:<INPUT TYPE=INPUT NAME="FirstName"><BR>
Last Name: <INPUT TYPE=INPUT NAME="LastName"><BR>
Address:   <INPUT TYPE=INPUT NAME="Address"><BR>
City:      <INPUT TYPE=INPUT NAME="City"><BR>
State:     <INPUT TYPE=INPUT NAME="State"><BR>
```

```
Zip:        <INPUT TYPE=INPUT NAME="Zip"><BR>
Phone #:    <INPUT TYPE=INPUT NAME="Phone"><BR>
E-mail.     <INPUT TYPE=INPUT NAME="E-mail"><BR>
</PRE>
<HR SIZE=5>
<INPUT TYPE=SUBMIT VALUE="Enter Information">
</FORM>
</BODY>
</HTML>
```

إنشاء شفرة عينة :

لمساعدتنا فى طريقنا لإنشاء خدمات WinCGI back - end فى بيسيك المرئية فقد ضمنت عينة شفرة لنا للنظر فيها . هذه العينة هى مساعد شكل (Form help-er) ، والذى يأخذ مستخدمى البيانات الذين أدخلوا فى شكل ويرسلوها بالبريد لشخص واحد أو أكثر، ويحفظها فى ملف ، أو كليهما ، ويمكننا تجهيزها لتعمل بأى شكل - (الأماكن التى تجهز موجودة فى الشفرة) . لإنشاء شفرة العينة الموصوفة ، نتبع هذه الخطوات:

١ - نبدأ بفتح بيئة بيسيك المرئية . نتأكد من تضمين CGI. BAS فى مشروعنا كما نحتاجه خلال الشفرة .

٢ - أولاً ، يجب أن نعلن المتغير الضرورى لإنهاء العمل لمعالجة الشكل الذى تم تقديمه . أمر Option Explicit يخبر بيسيك المرئية للتأكد أن كل المتغيرات قد تم إعلانها بالترتيب الذى سيستعمل .

```
' ****************************************************************
' Global Variable Declarations Section
****************************************************************
Option Explicit

' ****************************************************************
' Local (Module) Variable Declaration Section
' ****************************************************************
Const IsTRUE = 1
Const IsFalse = 0

' Used to store the names of the fields on the form.
Dim aFieldNames(500, 2) As String
```

```
' The following variables store information from the form.
Dim SaveToFile As Integer
Dim SaveFileName As String
Dim SendMail As Integer
Dim MailServer As String
Dim MailLogin As String
Dim MailToAddress As String
Dim FieldNames As String

Dim LastForm As form
Dim HTMLFileName As String

' ****************************************************************
' Constant Variable Declaration Section
'*****************************************************************
' These are variables which are used on the error pages.
Const' ServiceName = "Help Desk"
Const AuthorsE-mail = YourName@YourCompany.Com
Const AuthorsName = "Your Name"

' This is used when we need to output information for debugging.
' You can make this any filename you want to--however, the path
' must already exist.
Const DebugFile = "C:\Genie\HelpDesk.Dat"
```

٣ ـ الآن ، المتغيرات الضرورية فى مكانها ، يمكننا بدء برنامجنا . ونحن مطالبين أن يكون لدينا CGI - Main ( ) ، وإجراء Inter - Main ( ) فى شفرتنا CGI - Main () هو المسئول لأخذ تحكم من وحدة CGI. BAS متى أنهى متغيرات البدء . هذا هو حيث يبدأ البرنامج فعلياً . Inter Main( ) هو المسئول عن الاستجابة عن تنفيذ تفاعل بينى ، بمعنى آخر ، عندما يقوم مستخدم بتنفيذ البرنامج من رسالة أمر (Command Prompt) أو من مدير ملف مثل متبحر ويندوز.

```
' ******************************************************************
' This is the main routine of the Service. This procedure is called by *
' CGI.Bas:Main() after the CGI is initialized.                          *
' ******************************************************************
Sub CGI_Main ()
```

```
    If RequestMethod = "POST" Then
        EnterData                    ' Process the user's form
    Else
        ReturnForm         .         ' Return the Form for input.
    End If
End Sub                              ' End of CGI_Main()

' ************************************************************************
' This is the routine that runs when the user executes the program '
' from the Web server itself, interactively.                            '
' ************************************************************************
Sub Inter_Main ()
    MsgBox "This program only runs through the Web server."
End Sub
```

٤ ـ إذا وصل المستخدم للخدمة مثل a GET ، حينئذ يجب أن نرجع صفحة خطأ تخبرهم أن هذا ليس ممكناً : الروتين الفرعى التالى يؤدى هذا العمل .

ملاحظــة : GETs، POSTs هـى الطرق المستعملـة لدعـوة خدمـة طـرف بـه (back - end). A GET ترسل زوج Name = Value فى (Universal resource loccator :URL) ذاته. يتولد زوج Name = Value بحكم عبر HTML. طريقة POST كذلك ترسل أزواج Name = Value طريقةPOST تضع الأزواج فى جسم الرسالة ، وليس على الـ URL. وعند استقبال هذه الأزواج ، فإن خادم الشبكة (Web) يضعها فى ملف منفصل ، هذا الملف يوجد حيث تحصل شفرة CGI على المعلومة. معظم خدمات «الطرف ـ الخلفى» (back - end) هذه الأيام تدعى مع POSTs.

```
***********************************************************************
This subroutine is used to return a form to the user for input. Once *
the form is complete, the service runs again, calling EnterData().   *
***********************************************************************
Sub ReturnForm ()
   On Error GoTo ReturnFormError

  send ("Content-type: text/HTML")
  send ("")
  send ("<HTML>")
  send ("<Font Size=+3>")
  send ("This Service does not allow a 'GET' to be performed on it.")
  send ("You may only perform a 'POST' on this service.")
  send ("</Font>")
```

```
    send ("</HTML>")
    Exit Sub

ReturnFormError:
    send ("Content-type: text/HTML")
    send ("")
    send ("<HTML><TITLE>VB Error!!</TITLE>")
    send ("<H1>A Visual Basic error has occurred!</H1><HR>")
    send ("<I>Please note the following, and e-mail the author.</I>")
    send ("")
    send ("Service's Name:          " + ServiceName)
    send ("Subroutine :             ReturnForm()")
    send ("E-mail Service Author: <A HREF=""mailto:" + AuthorsE-mail + """>")
    send (AuthorsName + "</A></HTML>")
End Sub                                    ' End of ReturnForm()
```

٥ ـ إذا وصل المستعمل للخدمة بنجاح خلال شكل ، حينئذ يمكننا البدء بمعالجة معلومة الشكل . وفيما يلي الروتين الفرعي الذى يؤدى هذا العمل.

```
'**************************************************************************
' This subroutine is used to retrieve the data from the form and process *
' it.                                                                     *
'
'**************************************************************************
Sub EnterData ()
    On Error GoTo EnterDataError
    Dim sFieldNamesTemp As String
    Dim sSendMail As String
    Dim sSaveDatabase As String
    Dim sDatabaseFile As String
    Dim iFileNum As Integer
    Dim sOutput As Variant
    Dim sTempString As String
    Dim i As Integer
    Dim j As Integer
    Dim NumFields As Integer

      First, let's get the information from the hidden fields on the form.

   sFieldNamesTemp = GetSmallField("Field_Names")
   sSendMail = GetSmallField("Mail_Information")
   sSaveDatabase = GetSmallField("Save_To_File")

   ' Retrieve the names of the fields from the hidden form fields.
   i = 1
   j = 1
   Do
       While (Mid(sFieldNamesTemp, i, 1) <> "~") And _
               (i <= Len(sFieldNamesTemp))
           sTempString = sTempString & Mid(sFieldNamesTemp, i, 1)
           i = i + 1
```

```
    Wend
    i = i + 1
    aFieldNames(j, 1) = sTempString
    sTempString = ""
    j = j + 1
Loop Until i >= Len(sFieldNamesTemp) + 1
NumFields = j - 1

' Now retrieve the data from those fields on the form.
For i = 1 To (j - 1)
    aFieldNames(i, 2) = GetSmallField(aFieldNames(i, 1))
Next I

' If the user is going to send the information using e-mail, you first
' create the mail file. Then you send the mail using the hidden
' information on the form.
If UCase(sSendMail) = "YES" Then
    Create_Mail_File (NumFields)
    Send_Mail
End If

' If the user page is going to be stored into a database, you must output
' the information to a comma delimited text file.
If UCase(sSaveDatabase) = "YES" Then
    sDatabaseFile = GetSmallField("Data_File")
    sOutput = ""
    iFileNum = FreeFile
    Open sDatabaseFile For Append Access Write As #iFileNum
    For i = 1 To (j - 2)
    sOutput = sOutput & """" & aFieldNames(i, 2) & ""","
    Next I
    sOutput = sOutput & """" & aFieldNames(j - 1, 2) & """"
    Print #iFileNum, sOutput
    Close #iFileNum
End If

    ' Send those who submitted a form a response telling them that their
    ' information was processed successfully.
    SuccessFrm
 Exit Sub
 EnterDataError:
    send ("Content-type: text/HTML")
    send ("")
    send ("<HTML><TITLE>VB Error!!</TITLE>")
    send ("<H1>A Visual Basic error has occurred!</H2><HR>")
    send ("<I>Please note the following, and e-mail the author.</I><BR>")
    send ("<BR>")
    send ("Service's Name:         " + ServiceName + "<BR>")
    send ("Subroutine :            EnterData()" + "<BR>")
    send ("E-mail Service Author: <A HREF=""mailto:" + AuthorsE-mail + """>")
    send (AuthorsName + "</A></HTML>")
    End
 End Sub
```

٦ ـ فى الروتين الفرعى ، نحن ننشىء الملف الذى نريد إرساله بالبريد

الالكترونى للعنوان المختزن فى الحقل المختفى على الشكل (Form) . يحتوى الملف كل من الحقول على الشكل ، والبيانات التى تحتويها ، والتاريخ والوقت التى تم فيها تقديم الأشكال .

```
Sub Create_Mail_File (NumFields As Integer)
    On Error GoTo Create_Mail_FileError
    Dim iFileNum As Integer                    ' Stores next available File#
    Dim i As Integer
    iFileNum = FreeFile                        ' Get the next FileNumber
    Open ("C:\Genie\helpdesk.txt") For Output Access Write As iFileNum
```

ملاحظـــة : يمكننا تغيير إسم ملف النص الذى نكتب له بتعديل المسار وإسم الملف المعطى فى المثال السابق . نحن نحتاج فقط أن نتأكد أن المسار موجود.

```
Print #iFileNum, "The following information was sent by a _
                 user from the Web site..."
 Print #iFileNum, " "
 Print #iFileNum, "Date: " & Date
 Print #iFileNum, "Time: " & Format(Now, "hh:mm:ss")
 Print #iFileNum, " "
 For i = 1 To NumFields
 Print #iFileNum, aFieldNames(i, 1) & " = " & aFieldNames(i, 2)
 Next I
 Close iFileNum

    Exit Sub        ' Exit Create_Mail_File()

 Create_Mail_FileError:
    send ("Content-type: text/HTML")
    send ("")
    send ("<HTML><TITLE>VB Error!!</TITLE>")
    send ("<H1>A Visual Basic error has occurred!</H2><HR>")
    send ("<I>Please note the following, and e-mail the author.</I><BR>")
    send ("")
    send ("Service's Name:        " + ServiceName + "<BR>")
    send ("Subroutine :           Create_Mail_File()<BR>")
    send ("E-mail Service Author: <A HREF=""mailto:" + AuthorsE-mail _
         + """>" AuthorsName + "</A></HTML>")
    End
 End Sub                              ' End of Create_Mail_File()
```

٧ ـ فى الروتين الفرعى التالى ، فنحن نرسل ملف البريد والذى قد أنشىء تواً

للعنوان المسترد من الشكل (Form) . ونفعل ذلك ، فنحن نستعمل برنامج بريد مجانى للجمهور (Public domain) والمسمى mail . وقد نكون قادرين على الحصول على Blat وإلا ، يمكننا استعمال تطبيق عميل بريد الكترونى والذى يدعم متغيرات خط الأمر.

```
' ********************************************************************
' This is used to send out the mail message to the approvers.
' ********************************************************************
Sub Send_Mail ()
    On Error Resume Next
    Dim intHandle As Integer
    Dim sMessage As String
    Dim sMailServer As String
    Dim sMailTo As String
    Dim sMailLogin As String

    sMailServer = GetSmallField("Mail_Server_Name")
    sMailTo = GetSmallField("Mail_To_Address")
    sMailLogin = GetSmallField("Mail_Server_Login")
    intHandle = Shell("c:\Supertcp\blat.exe -install " & _
```

ملاحظـــة : ندخل المسار الصحيح لتطبيق عميل البريد الالكترونى الذى نستعمله .

```
            sMailServer & " " & sMailLogin)
 If Err = False Then
    Do While True
            AppActivate "Blat"
            If Err Then
                Err = False
                Exit Do
            End If
        Loop
    End If
    sMessage = "c:\blat\blat.exe C:\genie\helpdesk.txt -t " & sMailTo
    intHandle = Shell(sMessage)
    Exit Sub        ' Exit Send_Mail()
Send_MailError:
    send ("Content-type: text/HTML")
    send ("")
    send ("<HTML><TITLE>VB Error!!</TITLE>")
    send ("<H1>A Visual Basic error has occurred!</H2><HR>")
    send ("<I>Please note the following, and e-mail the author.</I><BR>")
    send ("")
    send ("Service's Name:         " + ServiceName + "<BR>")
```

```
    send ("Subroutine :           Send_Mail()<BR>")
    send ("E-mail Service Author: <A HREF=""mailto:" + AuthorsE-mail _
        + """>" + AuthorsName + "</A></HTML>")
End Sub          ' End of Send_Mail()
```

٨ ـ الشفرة التالية تستعمل لخرج المعلومة التى تمر خلال (معلومة)info الصف (String) داخل ملف نص لأغراض تصحيح الأعطال (debugging) .

```
' ************************************************************************
' This subroutine is used to print out information while debugging the *
' service. All of the information is stored in "C:\debug.txt"
' ************************************************************************
Sub debugout (Info As String)
    Dim iFileNum As Integer              ' Stores next available File#
    iFileNum = FreeFile                  ' Get the next FileNumber
    Open DebugFile For Append Access Write As iFileNum
    Print #iFileNum, Info
    Close iFileNum
End Sub                                  ' End of Debugout()
```

٩ ـ يستعمل الروتين الفرعى المعتاد ليرسل من الخاضعين(From Submitters) استجابة لاخبارهم أن شكلهم قد تم معالجته ويشكرهم لوقتهم . مرة أخرى ، نحن نستعمل المعلومة المختزنة فى الحقول المختفية على الشكل ، لذلك يمكننا تجهيز العنوان ونص الاستجابة . كذلك ، ندخل وصلة مجهزة لتحريك المستخدمين لموقع آخر .

```
' ************************************************************************
' This subroutine is used to return a success page to the user when the   *
' input is complete, valid, and has been entered into the database.       *
' ************************************************************************
Sub SuccessFrm ()
    Dim sTitle As String
    Dim sText As String
    Dim sLink As String
    Dim sLinkText As String

    sText = GetSmallField("Response_Text")
    sTitle = GetSmallField("Response_Title")
    sLink = GetSmallField("Response_Link")
    sLinkText = GetSmallField("Response_Link_Text")
    send ("content-type: text/html")
    send ("")
    send ("<HTML>")
    If sTitle = "" Then
    send ("<TITLE>No Title</TITLE>")
    Else
```

```
    send ("<TITLE>" & sTitle & "</TITLE>")
    End If
    send ("<BODY BGCOLOR=#ffffff>")
    send ("<H1>Thank You!</H1>")
    send ("<HR SIZE=5>")
    send ("<FONTSIZE=+3><I>")
    If sText = "" Then
        send ("Thank you for your information.")
    Else
        send (sText)
    End If
    If sLink <> "" And sLinkText <> "" Then
        send ("<BR><A HREF=""" & sLink & """>" & sLinkText & "</A>")
    End If
    send ("</FONT>")
    send ("</BODY>")
    send ("</HTML>")
End Sub                                        End of SuccessFrm()
```

الآن لدينا ، بعض الأفكار عن كتابة بيسيك مرئية . يمكننا إما نسخ الشفرة مباشرة أو نستعملها كمرجع لاعطائنا فكرة عما يمكن أن نفعله بتطبيق back - end الخاص بنا.

الباب الثانى

ويـــــندوز

(Windows)

الباب الثانى
وينـــــدوز
(Windows)

ويندوز عبارة عن بيئة زمن تشغيل . ومتى فهمنا الأساس لهذه البيئة الخاصة يمكننا البدء فى بناء تطبيقات الرسوم المتحركة (animation) بيسيك المرئية (Vis-ual Basic) والتى تستفيد كلية بالخواص التى تقدمها ويندوز . فى نفس الوقت ، فإن وينـدوز تنفذ برنامجنا ، وبرنامجنا ينفذ ويندوز . وكما نرى فى شكل (٢ ـ ١) ، فإن النموذج (Paradigm) المؤسس على رسالة يجعل الشركاء المتعاونين لتطبيقنا وويندوز فى وقت التنفيذ .

كيفية تنفيذ ويندوز لبرنامجنا :

ويندوز عبارة عن حكم (refree) فى لعبة مع كثير من اللاعبين . تطبيقنا هو واحد من هؤلاء اللاعبين. وويندوز تقدم بيئة تنفيذ طبيعية لتطبيقنا بأداء ثلاثة خدمات . الأولى، ويندوز تحمل وتنفذ برنامجنا .

وهذا يسمى ترويج (launching) تطبيقنا . الثانية ، فإن ويندوز تدير ذاكرة الحاسب الآلى بحيث يكون لتطبيقنا (وأى تطبيقات ويندوز أخرى والتى تكون فى حالة عمل فى نفس الوقت) وصول للذاكرة وموارد النظام الأخرى التى يحتاجونها . ثالثاً، ويندوز تتناول معظم دخل المستوى / المنخفض الذى يحتاجها تطبيقنا.

الترويج والتطبيق :

عندما ينتقى المستخدم أيقونة تطبيقنا ، فإن ويندوز تضع جانباً بعض الذاكرة

لبرنامجنا . حينئذ، تحمل فى الشفرة المنفذة وتهيىء كومة حمل . فهى تحمل أدنى (down loads) بيانات تطبيقنا وأى موارد خاصة أخرى مطلوبة بواسطة برنامجنا فى وقت التنفيذ. هذه الموارد الخاصة متفردة لبيئة ويندوز : فهى تتضمن نظام قائمة، ومفاتيح معجل ، وصناديق حوار وصناديق رسائل ، وأيقونات (Icons) وتشكيلات أسطح عرض (bitmaps) . ولانخلط موارد ويندوز مع موارد النظام مثل الذاكرة ، ووصول القرص ولوحة المفاتيح ودخل الفأرة والوصول لشاشة العرض .

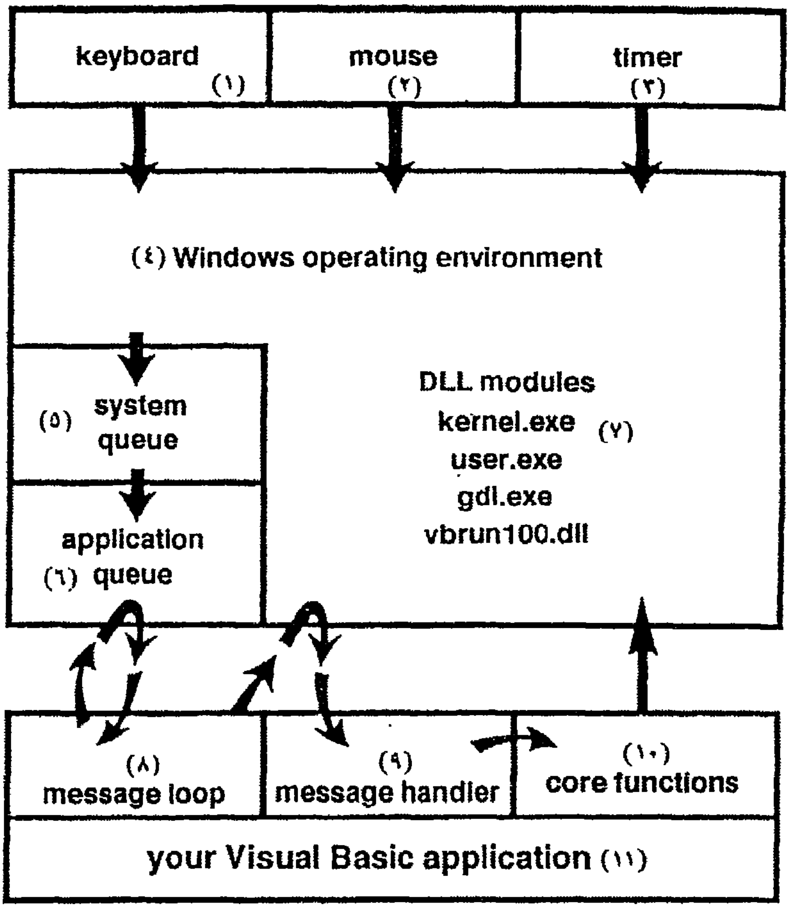
نظام بايتات فى ذاكرة عرض مثل :	bitmap
سطح عرض واقعى يمكن عمل رسومات عليه	

إدارة موارد النظام :

إذا كان يتم تنفيذ أكثر من تطبيق ويندوز واحد، فقد تكون الذاكرة محدودة . ويندوز تقدم مدير ذاكرة داخلى والذى يعمل فى الخلفية لتأكيد أن كل تطبيق له الذاكرة التى يحتاجها . أحياناً فإن هذا يعنى شفرة بديلة ، وبيانات ، وموارد خارج الذاكرة حتى تعمل مكان تطبيق آخر كل من هذه المكونات الثلاثة: الشفرة ، والبيانات والموارد تهيأ إلى وحدات تسمى قطع (Segments) ويندوز تبدل أو تلفظ القطع كلها عندما تدير الذاكرة .

إدارة الدخل :

ويندوز تعزل تطبيقنا من اعتماد الأجزاء الصلبة (hardware) باعطاء مجموعة غنية لخدمات الدخل والخرج الموجودة بالداخل . وهذا يجعل من الأسهل لنا أن نطور تطبيقات لأننا لا نحتاج لعمل أى سماحات لحاسبات آلية مختلفة ، والطابعات ، وهكذا .



شكل (٢-١) : البيسيك المرئية تتناول كل الفوقى الموجود فى حلقة الرسالة ومناول الرسالة حلقة االرسالة تستجلب رسائل واردة من نظام تشغيل ويندوز . مناول الرسالة يدعو الشفرة الوظيفية والتى لحقناها بفورمتنا .

(١) لوحة المفاتيح (٢) الفأرة (٣) الموقت (٤) بيئة تشغيل ويندوز (٥) صف النظام
(٦) صف التطبيق (٧) وحداتDll (٨) حلقة الرسالة (٩) مناول الرسالة (١٠) وظائف القلب
(١١) تطبيق بيسيك المرئية الخاصة بنا .

وكما نرى فى شكل (٢ ـ ١) ، فإن ويندوز تسجل (Poll) بصفة مستمرة لوحة المفاتيح ، والفأرة والوقت (Timer) . وعند حدوث حدث على واحدة من تلك الأدوات ، ويندوز تولد رسالة صحيحة وتضع الرسالة فى صف النظام (System

queue) . صف النظام هو قائمة من الرسائل دخول أول،خروج أول والتى تقابل رسائل تم استقبالها من لوحة المفاتيح والفأرة . وأحداث الوقت تختزن منفصلة.

ويندوز بارعة بدرجة لتعرف أى رسالة مخصصة لأى تطبيق ، حتى إذا كان أكثر من تطبيق واحد ينفذ. تطبيقات أخرى (والويندوز ذاتها) قد تبعث رسائل مباشرة لبرنامجنا فى وقت التنفيذ . الرسائل لتطبيقنا تختزن فى صف منفصل والذى تحفظه ويندوز بشكل واضح (expticitly) لبرنامجنا .

مثال رسالة :

فهم الرسائل هو مفتاح فهم برمجة ويندوز. تطبيقنا يتسلم الدخل باستجلاب الرسائل من الصف الذى تحفظه ويندوز لبرنامجنا . البيسيك المرئية تبنى أوتوماتيكياً شفرة صحيحة داخل تطبيقنا لتناول استجلاب وتوزيع الرسائل . كذلك ، يمكن لتطبيقنا أن يبعث رسائل لتطبيقات أخرى ولويندوز ذاتها . كذلك يمكنه استقبال رسائل من تطبيقات أخرى ومن ويندوز ذاتها مرة أخرى ، البيسيك المرئية تعتنى بكل التفاصيل لنا .

كل رسالة عبارة عن نواة (kernel) منتظمة من البيانات ، مختزنة فى هيكل بيانات داخل ويندوز يسمى MSG . هيكل MSG يحتوى كل ستة أعضاء :

wParam , WM , hwnd , Pt , Time , IParam

مقبض hwn window :

أول عضو يبين التطبيق الذى يجب أن يتسلم الرسالة . كل تطبيق ويندوز يجب أن يكون لديه ويندوز أساسى (يسمى ويندو عرض أيضاً) . مبرمجو البيسيك المرئية يرجعون لهذا الويندو كالشكل الأساسى ،الشكل الأساسى (main - form) ينفذ عادة كمنطقة مستطيلة على شاشة العرض . وكل ويندوز (أو شكل) محدودة بواسطة توكين (token) يدعى تداول (handle) يتذكر مثل hwnd . هذا العضو الأول لهيكل MSG يحتوى على تداول hwnd window .

محدد رسالة ـ الـ WM (Window message) :

العضو الثانى يحدد نوع الحدث الذى قد حدث . هذا العضو هو محدد الرسالة. وكل محدد رسالة يبدأ بحروف البدء (prefix) WM.

اختصار لرسالة ويندوز (Window message) . وإذا اختار المستخدم أمر من نظام قائمة تطبيقنا ، مثلاً ، فإن ويندوز سترسل رسالةWM... COMMAND لتطبيقنا.

I P aram , W Param

العضو الثالث والعضو الرابع يحتويان على متغيرات إضافية خاصة بالرسالة. فى حالة رسالة WM-- COMMAND ، فإن العضو الثالث لهيكلMSG يحتوى على كينونة بند القائمة والتى تم انتقائها بواسطة المستخدم . إذا تم تسلم رسالة له WM - KEYDOWN ، فإنها تبين أن المستخدم قد ضغط مفتاح على لوحة المفاتيح. فى هذه الحالة ، فإن العضو الثالث للرسالة يبين أى مفتاح قد تم ضربه. إذا تم تسلم رسالة WM - MOUSEMOVE ، فإنها تبين أن المستخدم قد حرك الفأرة . فى هذه الحالة ، العضو الرابع للرسالة يحتوى على الاحداثيات الجديدة لعلامة الفأرة . العضو الثالث للرسالة يدعى wParam ، والعضو الرابع يدعى IParam.

الوقت Pt :

العضوين الخامس والسادس لهيكل رسالة MSG يستعملان لمرات أقل . العضو الخامس، الوقت (t ime) يبين الوقت عندما وضعت الرسالة فى الصف (queue). العضو السادس pt يحتوى على الاحداثيات لعلامة الفأرة فى لحظة وضع الرسالة على الصف بواسطة ويندوز.

حلقة الرسالة :

تطبيقنا يستعمل رسائل باستاجلابها من الصف . وتوجد حلقة خاصة Special

(loop) (داخل برنامجنا بواسطة بيسيك المرئية حتى تقرأ الرسائل . فهى تسمى حلقة الرسالة (message loop) . ويتم إدارة حلقة الرسالة بواسطة وحدة شكل تطبيقنا .

صف : Queue

مجموعة متكررة من الإطارات والتى تتضمن دورة (حلقة) : loop

رسوم متحركة ؛ مثل دورة التنفيذ فى الرسوم المتحركة التقليدية للفيلم، أطوال الأفلام حيث يلصق طرف بطرف لتشكيل حلقة متكررة .

وإذا اكتشفنا تطبيقنا رسالة فى الصف (queue) فإنها تقفز خارج حلقة الرسالة لاداء النشاط (أياً كان) الذى برمجناه. حينئذ، يعود تحكم البرنامج لحلقة الرسالة ويبدأ فحص الصف لأول رسالة تالية واردة .

حلقة رسالة تطبيقنا تدعو وظيفة ويندوز بالداخل تسمى (Get Message) والتى تعطى تعليمات لويندوز لفحص الرسائل . وعند استدعاء Get Message . ويندوز تبحث أولاً فى صف التطبيق للرسائل ، وحينئذ تبحث فى صف النظام عن لوحة مفاتيح ورسائل فأرة ، ثم تفحص لترى إذا كانت أى رسائل موقت موجودة .

مناول الرسالة :

عندما يكتشف تطبيقنا رسالة واردة ويدعو إجراءات البيسيك المرئية (إجراءات Sub, Function) والتى بنيناها داخل برنامجنا . ويفعل ذلك بارسال رسالة خلال ويندوز لجزء خاص لبرنامجنا يسمى مناول الرسالة (message handler) . وكل تطبيق ويندوز يجب أن يكون له مناول رسالة . وهذا لأن ويندوز تصاحب كل ويندوز بمناولة رسالة . كذلك ، فإن مناول الرسالة يسمى إجراء ويندو أوتوماتيكيا، بيسيك المرئية تدخل شفرة صحيحة داخل تطبيقنا لتعمل كمناول رسالة .

ويندوز لا تسمح لتطبيقنا أن يدعو وظائف أخرى مباشرة . هذا لأن ويندوز بيئة أعمال متعددة تعاونية ، حيث يمكن تنفيذ أكثر من تطبيق واحد . التطبيقات الأخرى قد ترغب أن ترسل رسالة لمناول رسالة تطبيقنا ، حيث سيستقبل إهتمام مباشر من برنامجنا . إذا بعث الراسل رسالة للصف الذى تحفظه ويندوز لبرنامجنا ، يمكن وضع الرسالة خلف قائمة طويلة من الرسائل، وقد يكون برهة قبل أن يستجلبها برنامجنا من الصف. من الناحية الأخرى، إذا أرسلت الرسالة مباشرة لمناول رسالة تطبيقنا ، فلا يوجد تأخير وظائف ويندوز الموجودة بالداخل التى يمكن أن يدعوها تطبيقنا أن تمرر رسائل تسمى فى الحقيقة () Send Message ، و () Post Messager .

كيف ينفذ برنامجنا ويندوز :

كثير من العمليات الداخلية التى تتم بتطبيق تنفذ بدعوة الوظائف المبنية داخل إصدار صغير (retail) لويندوز . فى معظم الحالات، فإن بيسيك المرئية تعمل كوسيط بين شفرة مصدرنا ووظائف ويندوز المستدعاة . فى الحالات الأخرى، فقد نفضل أن ندعوا وظائف ويندوز مباشرة، ولكن بعمل ذلك فنحن نقلع عن فحص / متغير ممتد والذى تؤديه بيسيك المرئية لنا .

روتينات ويندوز المدعوة توجد فى مشترك برمجة تطبيق ويندوز (application programming interface : APT) . وعندما تستدعى هذه الوظائف ، فإن برنامجنا فعال، تنفيذ ويندوز . وعندما نستعمل وظائف متعددة، وعبارات (Statements) ، وخواص تزودها بيسيك المرئية ، وحينئذ فإن بيسيك المرئية تنفذ ويندوز بالإنابة عنا. وعندما ندعوا windows API مباشرة ، حينئذ فنحن ننفذ ويندوز.

يمكن لتطبيقنا أن يستعمل الـ API ليؤدى عديد من الأعمال ، ويمكننا استعمال API إما مباشرة أو خلال بيسيك المرئية . ويمكن لبرنامجنا أن يعيد ترتيب

أو يحرك ويندوز على شاشة العرض . يمكنه إعادة تشكيل نظم قائمة أثناء تنفيذ تطبيق . ويمكن لتطبيقنا أن ينسخ صورلـ قالب (Clipboard) الويندوز ، والتى منها يمكن استرجاع القالب إلى ويندوز تطبيقات أخرى ، والذى منها يمكن أن تلصق مع تطبيقات أخرى والتى تنفذ . يمكن لبرنامجنا أن يعطى تعليمات لمدير طبع ويندوز أن ينشىء نسخة صلبة لرسوماتنا على طابعة المستخدم . يمكننا استعمال API ويندوز لتوليد رسومات على ويندوز تطبيقنا ، وتصميمنا (رسوم متحركة animation).

قالب من الذاكرة الشاملة : Clipboard

يدار بواسطة ويندوز ليسمح للتطبيق أن يمرر . بيانات وصور لتطبيقات أخرى.

عرض سريع لصور منفصلة ، والتى تخدع العين البشرية animation

داخل الحركة المدركة . وتؤسس الرسوم المتحركة على خداع بصرى يسمى استبقاء الصورة (retention) وهو خاصية للعين البشرية .

مكتبات الوصلة الديناميكية :

الوظائف المدعوة (Callable) لويندز API محتواة خلال مكتبات زمن التشغيل تسمى مكتبات الوصلة ـ الديناميكية :(dynamic Link libraries DLLs). وتوجد ثلاثة DLLs مزودة بالإصدار الصغيرلويندز.

فهى مكتبة وصلة ديناميكية لويندز : KERNEL. EXE والتى تعطى موارد نظام مثل إدارة ذاكرة وإدارة مورد لتطبيق ويندوزفى زمن التشغيل.

gdi . exe, Kernel. exe, user. exe كما فى شكل (٢ ـ ١). البيسيك المرئية ذاتها تنفذ أيضاً مثل DLL . أى برنامج قابل للتنفيذ والذى ننشؤه بالبيسيك المرئية

يحتاج الوصول لمجموعة خاصة من الروتينات فى وقت التنفيذ (runtime) . هذه الروتينات توجد فى مكتبة الوصلة الديناميكية تسمى Vbran DLL.100. dll تعمل كمشترك بين برنامج بيسيك المرئية (Visual Basic) الخاص بنا وDLLs الثلاثة Windows API . وكمستخدم مرخص لـه لبيسيك المرئيـة ، لدينـا شهـادة مجانية (royalty - Free) لتوزيع نسخة لـ Vbran 100. dll بأى تطبيق ننشؤه باستعمال بيسيك المرئية .

Kernel . exe . dll تعطى وظائف لادارة ويندو عامة ولدعم بيئة ويندوز كلها . exe Dll المستخدم تقدم خدمات نظام مثل إدارة ذاكرة وإدارة وظائف متعددة. مشترك أداة الرسومات gdi.exe يقدم مجموعة من روتينات الرسومات ، وكثير منها مناسب جيداً للرسم المتحرك (animation) . برامج توضيح البيسيك المرئية فى الكتاب تستفيد من الوظائف المدعوة (Callable) الموجودة داخل آلة رسومات . gdi . exe

مكتبة وصلة ديناميكية لويندو :	GDI.EXE
والتى تعطى وظائف رسومات مستقلة خادعة لتطبيقات ويندو.	

بناء تطبيق بيسيك أساسى :

تطوير واختبار تطبيق بيسيك مرئية مثل استعمال قوالب بناء ، ونوظف مشترك الرسومات المتفاعل بينياً لبيسيك المرئية لاختيار وتجميع قوالب البناء هذه إلى نموذج أصلى (Prototype) تطبيق . حينئذ ، نلحق شفرتنا الأصلية للنموذج الأصلى حتى نحمل الأعمال التى يؤديها برنامجنا للمستخدم .

النموذج الأصلى :	Prototype
عبارة عن نموذج تجريبى لبرنامج لأغراض تخطيط / مشروع .	

● **قوالب بناء التطبيق :**

البيسيك المرئية تعطى قوالب بناء أساسية :

● أشكال (Forms)

● تحكمات

● خواص

● أحداث (events)

● طرق

● قوائم (menus)

*** الأشكال :**

الأشكال (Forms) هى ويندوز . وكل تطبيق ننشؤه باستعمال البيسيك المرئية يجب أن يكون له شكل أساسى والذى يسمى ويندو أساسى أيضاً ، أو ويندو عرض . البيسيك المرئية تنشىء أوتوماتيكياً شكل أساسى لتطبيقنا . الشكل هو نوع من الهدف (object) فى مشروع البيسيك المرئية للبرمجه .

*** التحكمات :**

التحكمات (Controls) عبارة عن أشياء رسومات والتى نضعها فى شكل حتى نصيد دخل مستخدم وخرج عرض . أمثلة تحكمات البيسيك المرئية هى أزرار الأمر (command)، وصناديق القائمة وصناديق النص ، وصناديق الصور . وكل تحكم كذلك ينجز مجموعة من الخواص المعروفة ، وأحداث وطرق مثل الأشكال، فإن التحكم هو نوع من الهدف فى نظام البيسيك المرئية لتطوير التطبيق.

*** الخواص :**

الخواص (Properties) هى خواص (attributes) لشكل أو تحكم . الخواص

تستعمل لتعريف الحجم واللون والمحتويات وموقع الشكل أو التحكم . وكذلك ، فإن الخواص يمكن استعمالها أيضاً لوصف تصرفات شكل أو تحكم ، سواء كان مرئياً أم مدعماً (enabled) . مثل عبارة :This Picture. Visible = True ستجعل صندوق الصورة مسمى This Picture visible

*** أحداث :**

الحدث (event) عبارة عن نشاط مثل طقطقة فأرة (mouse click) أو مشوار مفتاح والذى يتم التعرف عليه بشكل أو تحكم . يمكننا كتابة قالبنا للشفرة والذى سيدعى أوتوماتيكياً عندما يكتشف الشكل أو التحكم الحدث . قالب الشفرة الذى نزوده يدعى إجراء حدث. فمثلاً، إجراء حدث يسمى () - IDM Menultem - Click سيدعى أوتوماتيكياً عندما يطقطق المستخدم على بند القائمة المسمىIDM - Menultem. اجراءات الحدث هى دائماً إجراءات فرعية (Sub)، والتى لا ترجع قيمة. النوع الآخر للإجراء المدعم ببيسيك المرئية يسمى إجراء Function (وظيفة) والذى لايرجع قيمة.

*** الطرق :**

الطريقة (metbods) تشبه وظيفة البيسيك المرئية المعتادة أو العبارة (Statement)، فيما عدا أن الطريقة تعمل على الأهداف (objects). هدف البيسيك المرئية هو شكل أو تحكم . كذلك ، يمكن استعمال الطريقة بأهداف متخصصة مدعمة بالبيسيك المرئية، مثل الطابعة وقالب ذاكرة شاملة (Clip-board). فمثلاً ، استعمال العبارة Form I Hide يقحم طريقة الاختفاء (Hide) لازالة الشكل الأساسى من العرض. وهذا يماثل تهيئة خاصية المرئى إلى خطأ (False) .

*** القوائم :**

القوائم (menus) وقضبان قوائمها يمكن إنشاؤها باستعمال نافذة تصميم

قائمة بيسيك المرئية (Visual Basic Menu Design Window) . بنود القائمة المنفردة تسمى تحكمات القائمة . عندما تستعمل ويندو تصميم قائمة لتصميم نظام قائمة تطبيقنا ، نحدد إسم تحكم لكل بند قائمة. إسم التحكم هذا يستعمل بواسطة البيسك المرئية لدعوة إجراء الحدث الذى نزوده . فمثلاً، إذا حددنا IDM- This Item كاسم تحكم ، حينئذ فإن بيسيك المرئية ستدعو إجراء حدث يسمى ( ) IDM-This Item - Clck فى كل مرة طقطق المستعمل على بند القائمة . ومثل المبرمج ، فنحن مسئولون عن إعطاء الشفرة لإجراء الحدث المسمى IDM - ThisItem - Click () . البيسيك المرئية تعطى قالب (Template) شفرة يحتوى على العبارة الأولى والعبارة الأخيرة فقط لإجراء الحدث ، والنتيجة ، لن يحدث أى عمل حتى يضيف عباراتنا بين End Sub،Sub المزودة بواسطة بيسيك المرئية .

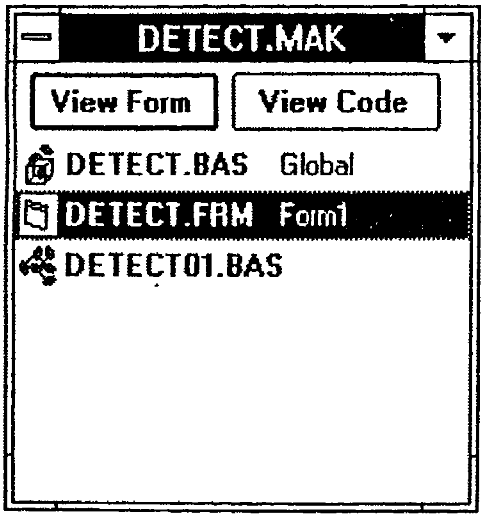
قالب أو عارضة :	Template

شكل الكترونى مصنع لتطبيقات برامج محددة ، مثل إنشاء صحيفة غطاء فاكس.

العمل ببيسيك المرئية :

العمل ببيسيك المرئية يعنى العمل بأربعة أنواع من الويندوز المزودة بواسطة بيسيك ذاتها:

- ويندوز المشروع
- ويندو شكل زمن ـ التصميم
- ويندو تصميم القائمة
- ويندوز الشفرة



شكل (٢-٢) ويندو مشروع البيسيك المرئية .

كل تطبيق عينة فى الكتاب يبنى من مجموعة من الشكل الشامل ووحدات وظيفية.

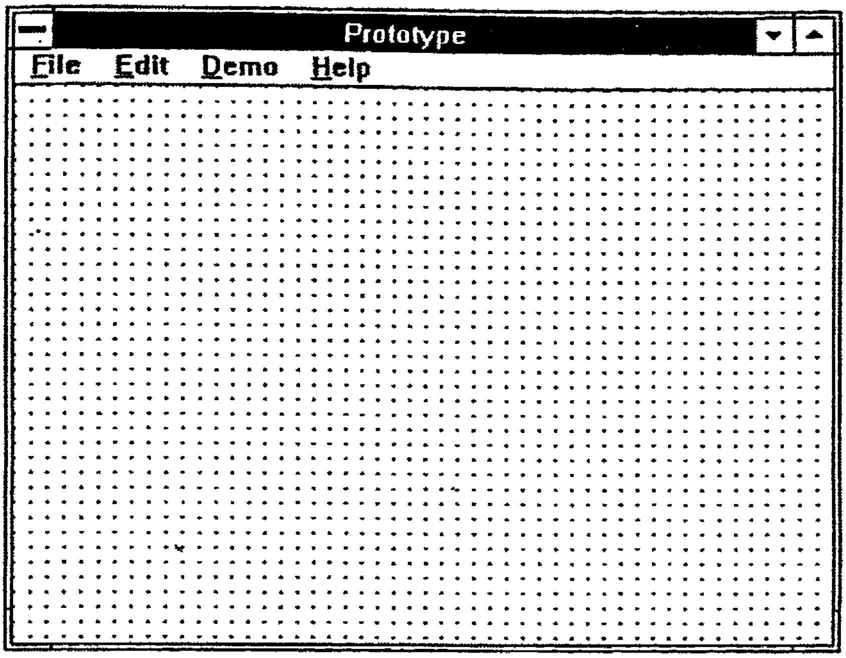
استعمال ويندو المشروع :

ويندو المشروع يحتوى على قائمة لكل الأشكال والوحدات والتى تعمل تطبيقنا. ويندو المشروع لتطبيق بيسيك مرئية معتادة مبين فى شكل (٢ ـ ٢).

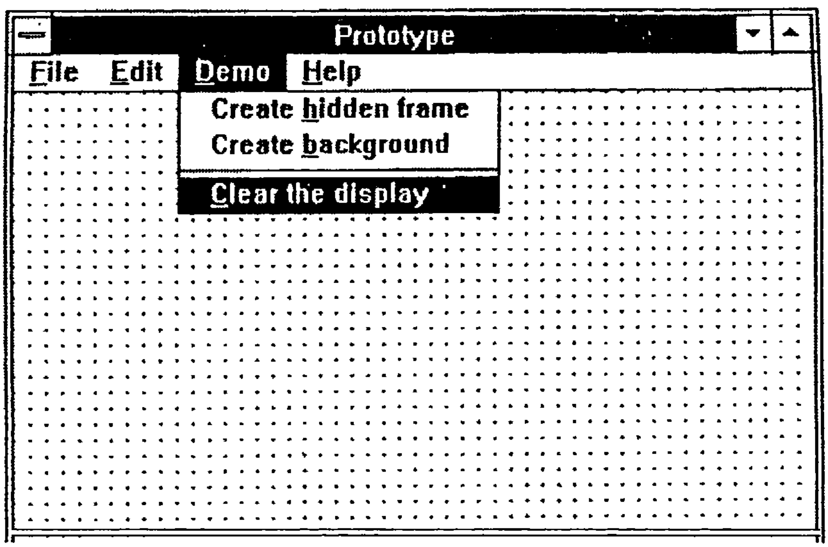
استعمال ويندو شكل وقت ـ التصميم:

ويندو شكل وقت ـ التصميم (The design - time Form window) هو تمثيل بالرسم لنافذة عرض تطبيقنا . ونحن نضع تحكمات مثل الموقتات وصناديق صور أو ويندو شكل وقت ـ التصميم أثناء بناء تطبيقنا . ويندو شكل وقت ـ التصميم لدورة برمجة بيسيك مرئية معتادة مبينة فى شكل (٢ ـ ٣) . نموذج الشبكة مزودة بواسطة البيسيك المرئية لمساعدتنا على وضع تحكمات على ويندو شكل وقت ـ تصميم.

ونستعمل ويندو تصميم قائمة (Menu Design Window) لبناء نظام ويندو نموذج أصلى (Prototype) لتطبيقنا على ويندو شكل وقت ـ تصميم ـ قائمة نموذج أصلى (Prototype) فى ويندو شكل وقت تصميم لدورة برمجة بيسيك مرئية معتادة مبين فى شكل (٢ـ٤).



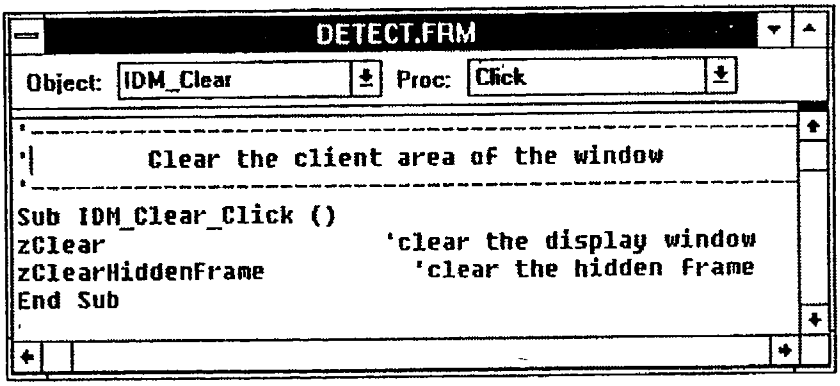
شكل (٢-٣) ويندو شكل وقت تصميم بيسيك مرئية



شكل (٢-٤) كل تطبيق عينة فى الكتاب يلتصق بنظام قائمة متماسكة (consistent) ، لذلك يمكننا التركيز على الرسومات التى يتم شرحها

ربط شفرتنا :

شفرة الشكل (Form code) هى شفرة إجراءات أحداث وأساسية فى ملف the Frm حيث أن الشكل (Form) ذاته معرف . وعندما نكتب شفرة ربط (attaching Code) شكل محدد أو تحكم . ويندو المعتادة مبينة فى شكل (٢ - ٥). الاجراء الفرعى Sub المسمى IDM - Clear Click () يستدعى أوتوماتيكيا عندما ينتقى مستخدم تطبيقنا بند القائمة المسمى IDM - Clear . الشفرة الوظيفيه (Functional Code) ، متضمنة إجراءات Sub ، Function والتى نزودها لأداء أعمال عامة . الشفرة الوظيفية تختزن عادة فى وحدات bas . منفصلة عن وحدة Frm . الاجراءات فى وحدة Frm . يمكنها استدعاء إجراءات فى وحدات bas . ، ولكن الشفرة فى وحدات bas . لا يمكنها استدعاء اجراءات فى وحدة Frm .

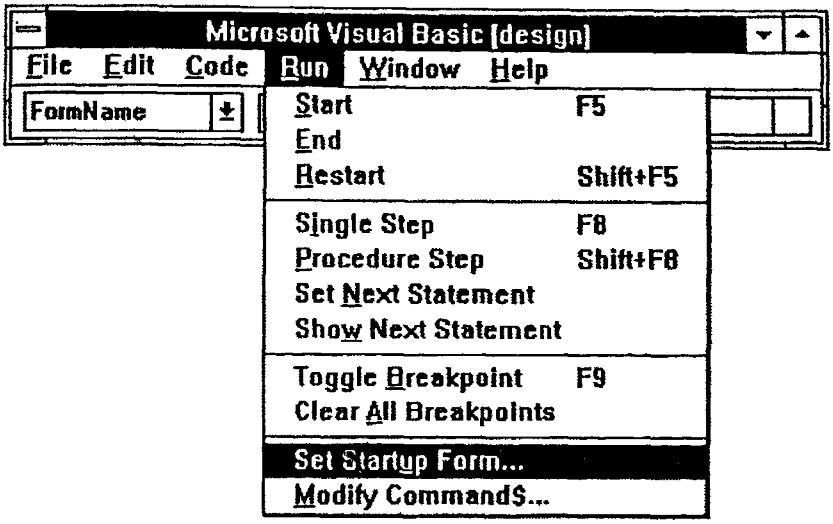


شكل (٢-٥) : ويندو شفرة بيسيك المرئية . المبين هنا شفرة، ويتم استدعاؤها عندما يستعمل المستخدم الفأرة للطقطقة على بند قائمة يسمى IDM - Clear

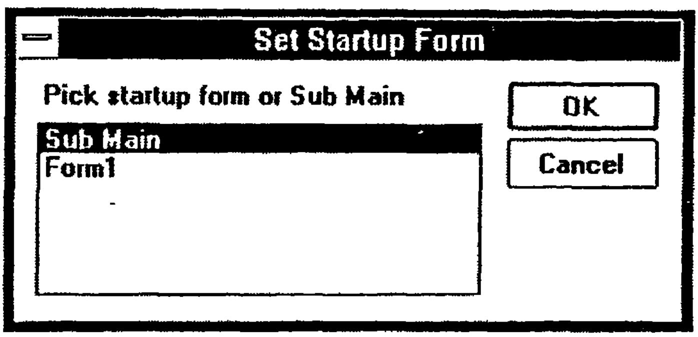
عمل نماذج أصلية بالبيسيك المرئية

أثناء التجميع ، واختبار ، وإعادة إختيار تطبيقنا ، فنحن نعمل النموذج الأصلى. (Prototyping) البيسيك المرئية ، تجعل من السهل لنا أن ننفذ تطبيقنا والرجوع بسرعة لبيئة البيسيك المرئية. قائمة Run لبيئة برمجة للتفاعل البينى لبيسيك المرئية

مبينة فى شكل (٢ ـ ٦) . أنتقاء Run يجعل تطبيق نموذجنا الأصلى(Prototype) أن يطلق تحت تهيئة تحكم البيسيك المرئية . انتقاء شكل بدء تهيئة.. Set Startup) Form ..) يجعل صندوق الحوار فى شكل(٢ ـ ٧) يعرض ، والذى يحدد نقطة بدء التنفيذ لتطبيقنا .



شكل (٢-٦) : قائمة تنفيذ وقت تصميم (design - time) . كل تطبيق عينة فى الكتاب يستعمل وحدة بدء مجهزة، والتى يجب أن تنشط بانتقاء بند قائمة شكل قائمة(Set Startup Form menu)



شكل (٢-٧) : صندوق حوار بيسيك المرئية . تهيئة شكل البدء (Set startup form) تسمح لنا أن نحدد أى وحدة تستدعى أولاً عندما يبدأ تنفيذ تطبيقنا .

استعمال ويندو تصميم القائمة :

قائمة ويندو المبينة فى شكل (٢ـ ٨) ، يستعمل لتنشيط ويندو تصميم القائمة (Menu design Window) المبين فى شكل (٢ ـ ٩) . ويمكننا استعمال ويندو (نافذة) تصميم القائمة لتحديد أسماء القائمة والتى تظهر على قضيب القائمة لتطبيقنا ، وكذلك بنود منفردة التى تظهر فى كل قائمة مخربة(pull - down) كما هو مبين بواسطة العينة فى شكل (٢ ـ ٩) ، فإن خاصية (Caption) هى الاسم المتعلق بالنص (Textual) لبند القائمة والذى سيظهر فى القائمة. خاصية CtiName هى إسم التحكم الذى ستدعوه بيسيك المرئية عند حدوث حدث ـ شكل (٢ ـ ٥).

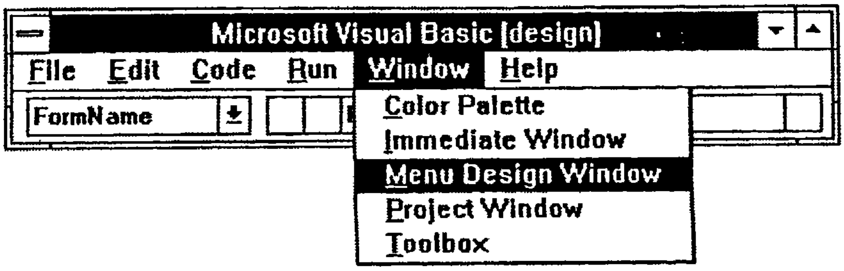
قضيب العنوان لويندو الأساسى : Caption bar
لتطبيق ويندو . ويحتوى عادة إسم التطبيق .

كذلك ، يمكننا استعمال نافذة تصميم القائمة (Menu Design Window) لإضافة مفاتيح إختصار (Shortcut) لنظام قائمتنا . يمكن ضغط مفاتيح اختصار (Shortcut) بواسطة المستخدم فى أى وقت أثناء تنفيذ برنامج، وأمر قائمته المقابل سيتم تنشيطه.

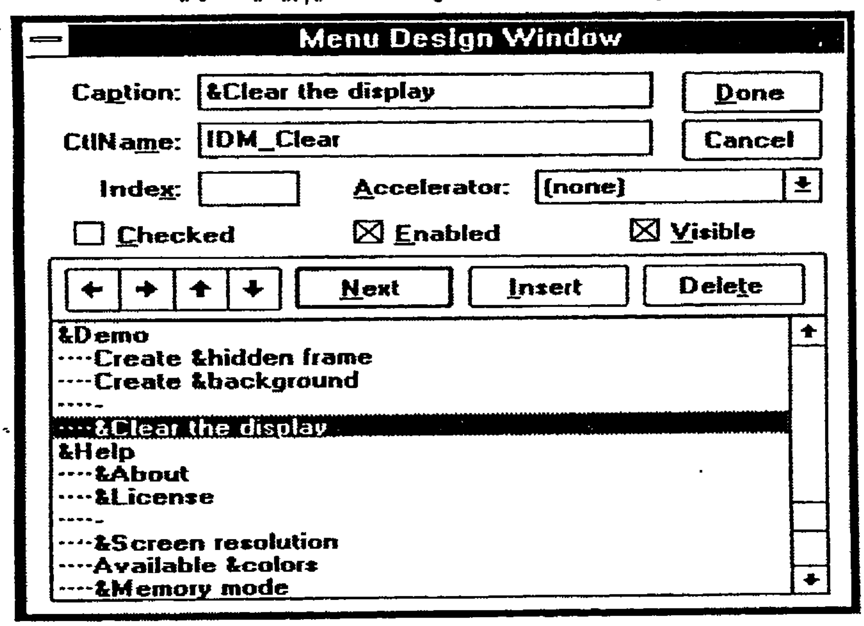
فمثلاً ، فإن العينات الموضحة فى هذا الكتاب تستعمل مفتاح F1 كاختصار لتنشيط صندوقHelp وقت التنفيذ. المفاتيح المختصرة تسمى مفاتيح المعجل(ac-celerator keys) بواسطة مبرمجى ويندوز.

كذلك ، فإن مفاتيح الوصول (access) مدعمة بواسطة نافذة تصميم القائمة (Menu Design Window). فمثلاً إذا أدخلنا رمز & (و) (ampersand) كجزء منCaption لإسم قائمة ، مثل Thi & sMenu ، حينئذ يمكن للمستعمل تنشيط القائمة بضغط وحفظ مفتاح Alt لأسفل أثناء ضرب مفتاح S . إذا

استعملت رموز&، وفى عناوين (Captions) لبنود قائمة ، مثل& IDM Menu Item ، حينئذ ، فإن المستخدم يحتاج أن يضغط / فقط لتنشيط أمر IDM - Menu Item . لأن تطبيقات العينات خلال هذا الكتاب تستعمل مفاتيح وصول فى نظم قوائمها ، يمكننا استعمال إما فأرة أو لوحة المفاتيح لتشغيل البرامج . مفاتيح الوصول (access keys) تسمى مفاتيح ذاكرة (mnemonic) أيضاً بواسطة مبرمجى ويندوز.



شكل (٢-٨) : قائمة نافذة وقت / تصميم بيسيك المرئية



شكل (٢-٩) : نافذة تصميم القائمة مغطاة بواسطة بيسيك المرئية . وكل تطبيق عينة فى الكتاب يستعمل نظام قائمة متماسكة (Consistent) والتى يجب أن نبنيها عند وقت / التصميم

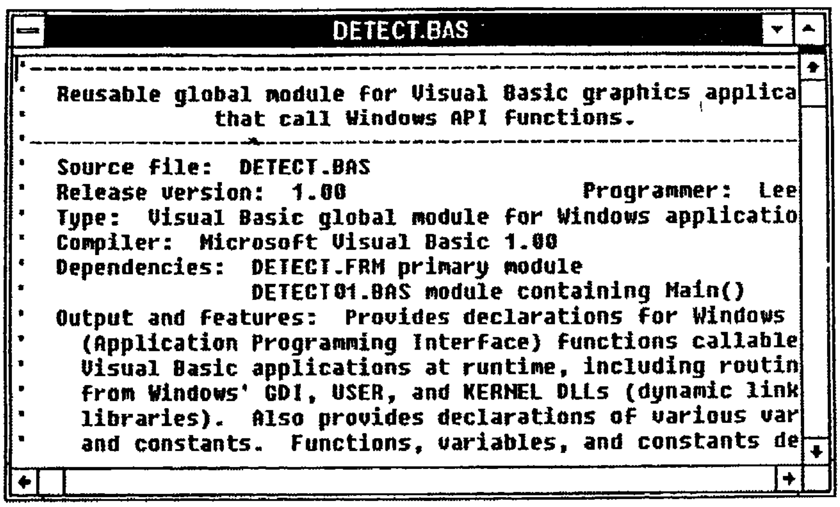
البرمجة متعددة الوحدات ببيسيك المرئية :

تطوير تطبيق بيسيك المرئية عبارة عن برمجة متعددة الوحدات (multimodule) ثلاثة أنواع من الوحدات تستعمل فى تطبيق بيسيك مرئية معتادة :

- الوحدة الشاملة : (THe global module)
- وحدة الشكل : (The Form module)
- الوحدة (أو الوحدات) الوظيفية : (The Functional module)

استعمال الوحدات الشاملة :

الوحدة الشاملة لتطبيقنا تحتوى على بيانات (declarations) ثابتة ، وبيانات متغيرة ، وبيانات إجراءات والتى تظهر فى وحدات التطبيق. وكل تطبيق بيسيك مرئية له وحدة شاملة . تطبيقات العينة فى هذا الكتاب تستعمل الوحدة الشاملة لإعلان إجراءات فى مشترك برمجة تطبيق ويندوز (Application Programming Interface: API) والتى يمكن استدعاؤه بشفرتنا .



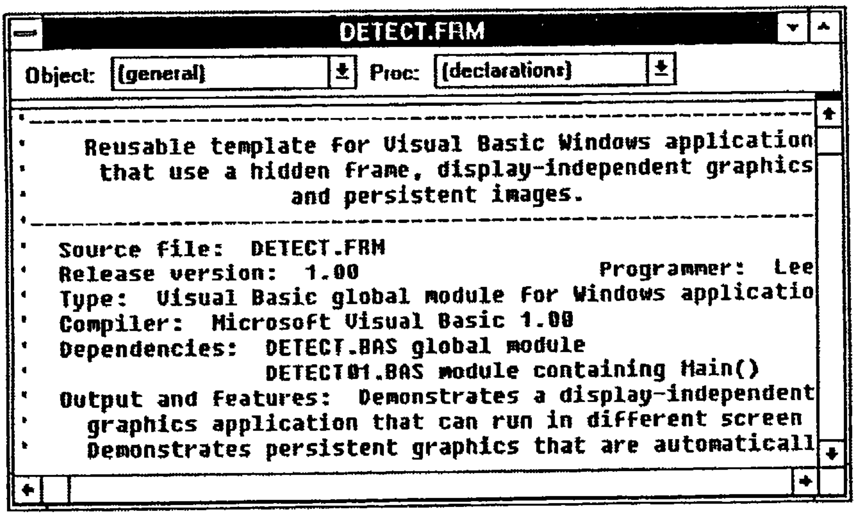

```
'------------------------------------------------------------
'  Reusable global module for Visual Basic graphics applica
'         that call Windows API functions.
'------------------------------------------------------------
'  Source file:  DETECT.BAS
'  Release version:  1.00                    Programmer:  Lee
'  Type:  Visual Basic global module for Windows applicatio
'  Compiler:  Microsoft Visual Basic 1.00
'  Dependencies:  DETECT.FRM primary module
'                 DETECT01.BAS module containing Main()
'  Output and Features:  Provides declarations for Windows
'    (Application Programming Interface) functions callable
'    Visual Basic applications at runtime, including routin
'    from Windows' GDI, USER, and KERNEL DLLs (dynamic link
'    libraries).  Also provides declarations of various var
'    and constants.  Functions, variables, and constants de
```

شكل (٢-١٠) : ويندو شفرة بيسيك مرئية . والمبين هنا عبارة عن وحدة شاملة معتادة تحتوى على ثوابت ومتغيرات مرئية لكل الوحدات فى التطبيق ، متضمنة إعلانات تجعل من الممكن دعوة وظائف رسومات GDI .

ويندو الشفرة لوحدة شاملة معتادة مبينة فى شكل (٢ ـ ١٠). وتحتوى الوحدة الشاملة على ثوابت ومتغيرات مرئية خلال تطبيقنا ، متضمنة إعلانات تجعل من الممكن لتطبيقنا أن يدعو وظائف مباشرة لمشترك أداة رسومات ويندوز (Windows Graphics Device Interface: GDI) . هذا الوصول لـ GDI يعطى الدعم (Underpinnings) لقدرات رسوم متحركة متقدمة لبرامج demo . الوحدة الشاملة تستعمل bas Suffix .

استعمال وحدات شكل :

ويندو الشفرة لوحدة شكل معتادة مبينة فى شكل (٢ ـ ١١). وحدة الشكل (Form module) تحتوى على شفرة مصدر إجراء / حدث . وحدات الشكل تحتوى أيضاً على خواص الشكل الأساسى ونظام قائمة لتطبيقنا . وتختزن وحدة الشكل فى نسق خاص يمكن قراءته بواسطة بيسيك المرئية. وحدات الشكل تستعمل إمتداد إسم ملف . قبل أن نتمكن من طبع قائمة لوحدة شكل ، يجب أن نستعمل بيسيك المرئية لتحويلها الملف .



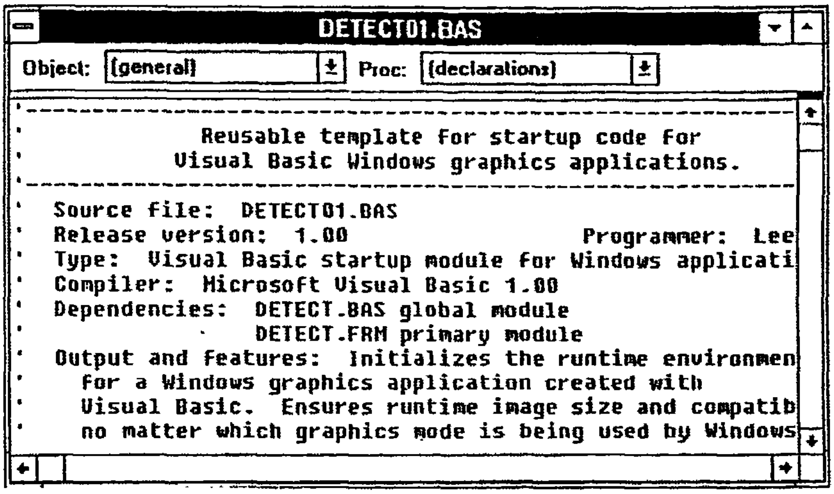
شكل (٢-١١) : ويندو شفرة بيسيك مرئية . المبين هنا هو وحدة شكل معتادة، تحتوى على شفرة إجراء تستدعى كاستجابة لأحداث مثل طقطقات فأرة (mouse clicks) .

استعمال وحدات وظيفية :

ويندو الشفرة لوحدة وظيفية معتادة مبينة فى شكل (٢ ـ ١٢). وتحتوى الوحدة الوظيفية على روتينات (إجراءات Function ، Sub) والتى يمكن لاجراءات الحدث فى وحدة الشكل الخاصة بنا أن تدعوها لأداء أعمال محددة . ويلاحظ أن الإجراءات فى وحدة وظيفية لايمكنها دعوة إجراءات فى وحدة الشكل. تطبيق بيسيك مرئية قد لا يكون له شيء ، أو له واحد أو إثنين أو ثلاثة أو أكثر للوحدات الوظيفية . معظم عينات التطبيقات فى هذا الكتاب لها وحدتين وظيفيتين .

الوحدة الوظيفية الأولى تحتوى على شفرة بدء تستعمل لبدء متغيرات شاملة [نرجع مرة أخرى لشكل (٢ ـ ٦) ، شكل (٢ ـ ٧)]. الوحدة الوظيفية الثانية تحتوى على إجراءات عامة . الوحدات الوظيفية تستعمل إمتداد إسم ملف bas . ومثل وحدة الشكل (Form) ، ويجب تحويل الوحدة الوظيفية لملف tex. قبل طبع قائمة برنامج .

وعندما تجمع مع خواص برمجة المستوى / العالى المزودة بواسطة لغة بيسيك المرئية ، فإن هذه الأنواع الثلاثة من الوحدات (الشاملة والشكل والوظيفية) تعطينا تنوع غير منتظر فى إنشاء تطبيقات رسومات لويندوز ، حيث أن تطبيق العينة فى الباب التالى سيشرح ذلك .



شكل (٢-١٢) : ويندو شفرة بيسيك مرئية . المبين هنا هو وحدة وظيفية معتادة (فى هذه الحالة ، فإن وحدة البدء تدعى عندما يبدأ تطبيقنا فى العمل أولاً).

الباب الثالث

الرســـومات

(Graphics)

الباب الثالث
الرسومـــات
(Graphics)

ميكروسوفت ويندوز تعطى رسومات بها تسمى مشترك أداة رسومات(graphics device interface : GDI). وظائف GDI يمكن دعوتها بواسطة تطبيقات بيسيك المرئية فى وقت التشغيل . هذه الوظائف المستدعاة موجودة فى gdi . exe ، وهى واحدة من مكتبات الوصلة / الديناميكية المضمنة مع كل إصدار مجزأ (retail) لويندوز .

GDI تعطى روتينات رسومات قوية والأقرب للكمال (Optimized) لويندوز . بالإضافة للخط العيارى (Standardline)، وقطع ناقصellipse وروتينات الملء (Fill)، فإن GDI تقدم مختارات لوظائف الارتجاف (dithering)، bitblt والقص . هذه الأنواع الثلاثة من الوظائف تناسب جيداً لعمل الرسومات المتقدمة . وظائف bitbltمفيدة خاصة لبناء تتابعات رسوم متحركة فى تطبيقات بيسيك المرئية .

إختصار لنقل قالب أرقام ثنائية :	. bitblt
(bit block transfer) كذلك ، يسمى رسومات قالب ومصفوفة رسومات .	
كسوة الرقم الثنائى أو عمل نماذج :	dithering
لعناصر الصورة المستعملة لتنفيذ ظل أو مشروع ألوان.وويندوز تستعمل هذا الارتجاف لتحاكى ألوان بعد الانتقاء المحدود المتوفر فى طور VGA ١٦ لون : إرتجاف .	

:GDI

GDI عبارة عن آلة رسومات ضيقة / ويندوز . ويمكن دعوة وظائفGDI بواسطة أى تطبيق ينفذ فى بيئة ويندوز . ويمكن لـ GDI أن توجه خرج رسوماتها للشاشة والذاكرة والطابعة وملف القرص وأدوات أخرى . وبسبب هذه المرونة للخرج، يجب أن نعطى بيئة (Context) فى كل مرة ندعوا وظيفة فىGDI . المحيط: (Context) هو مجموعة من الافتراضات والتى كلانا و GDI نعملها عن أداة الخرج الحالى . الكلمة أداة (device) لها معنى خاص لـ GDI .

وسيلة لادارة الذاكرة . محيط (بيئة):.	Context

الأدوات : (Devices)

الأداة هى نقطة بؤرية للدخل والخرج . لوحة المفاتيح والفأرة والموقت هى أدوات دخل . الشاشة والطابعة وقلم الرسم (Plotter) ، وتشغيل القرص (disk drive) والموديم (modem) هى أدوات خرج. كذلك ، فإن أدوات الخرج يمكن أن تتضمن أدوات محاكيه (Simulated) فى ذاكرة ، مثل bitmaps وما بعد الملفات (metafiles) . a bitmap عبارة عن سطح رسم محاكى . ما بعد عبارة عن مجموعة استدعاءات وظائفGDI والتى يمكن استعمالها لاختزان وسائل إعادة إنشاء صورة .

وقبل دعوة وظيفة رسومات ، يجب أن نعطى معلومة (محيط / أداة) /device context للـ GDI ويجب أن نخبر الـ GDI أين وكيف نكتب خرج البيانات . ونحن نؤدى ذلك بإنشاء device / context (محيط / أداة).

وسائل إدارة الذاكرة (Contexts) / الاداة :

(Context) الأداة يمثل أداة خرج ومشغل أداته . مشغل الأداة (device driver) عبارة عن برنامج والذى يعالج أداة المكونات الصلبة مباشرة . الطابعة لها مشغل أداة . موائمات الرسومات لها مشغل أداة . الفأرة لها مشغل أداة .

إذا كنا ننفذ ويندوز على حاسب مزود / VGA ، فإن البرنامج الذى يعالج ذاكرة العرض لموائم الرسومات موجود فى مشغل يسمى Vga . drv. هذا المشغل يتخلف عادة على دليل C: \ windows \ system على قرصنا الصلب . ويندوز تحمل مشغل الأداة هـذا داخل ذاكـرة عند البـدء ، وعندمـا يدعـو تطبيقنـا الـ GDI لانشـاء رمـوز (imagery) رسومات على الشاشة ، الـ GDI يدعو روتينات فى Vga . drv لمعالجة موائمات رسومات VGA . موائمVGA هو الأداة ، و Vga . drv هوالمشغل .

محيط أداة (device / Contexts) محددة بواسطةGDI باستعمال ما يسمى المقابض (handels) وسائل إدارة الذاكرة / الأداة، لكنها وصف عديـد مـن أدوات الخرج. محيط (Context) العرض هو نوع واحد من محيط/ أداة / device) .context)

مصفوفة رسومات الفيديو	VGA : Video graphics array

محيطات (Context) العرض :

محيط العرض يحدد نافذة على الشاشة . بالأخص ، فان Context العرض يصف كيف تكتب مخارج الرسومات لمنطقة العميل لنافذة محددة. منطقة العميل هى الجزء الداخلى أوسطح رسم ، للنافذة (لا يتضمن قضيب العنوان (Caption ، وقضيب القائمة والحدود "borders") . مبرمجو ويندوز يستعملون الاختصار DC ، غالباً ليعنى إما محيط (Context) عرض أو محيط / أداة.

محيط العرض :

عندما نستعمل GDI لبدء محيط / عرض لنافذة ، فإن ويندوز تعيد تهيئة أصل الرسومات للركن العلوى الأيسر لمنطقة عمل ويندوز، وتعيد تهيئة مستطيل القص

لأبعاد منطقة العميل .. أصل الرسومات هو موقع عنصر الصورة الذى يوصف بالاحداثيات O،O . مستطيل القص هو منفذ رؤية (Viewport) للرسم.GDI سيقص كل الرسومات عند طرف منفذ الرؤية ولن يرسم خارج مستطيل القص . أصل الرسومات ومستطيل القص هما مجرد إثنين للخواص الافتراضية الكثيرة لمحيط / العرض.

شكل (٣-١) : نافذة عرض لتطبيق رسوماتنا
ستعرض هذه الخواص الافتراضية إلا إذا حددنا أخرى

Default Attributes (الخواص الافتراضية)	
لون أرضية خلفية	أبيض
طور أرضية خلفية	معتم
لون فرشاة	أبيض
لون قلم	أسود
لون نص	أسود
طاقم (Font)	طاقم النظام ، ونظام ومتناسب
منطقة القص	منطقة العميل
أصل DC	(O , O) الركن العلوى الأيسر، ومنطقة العميل
موضع القلم	O،O
موضع الفرشاة	O،O

الخواص الافتراضية : (default attributes)

تلخيص الخواص DC الافتراضية مبين فى شكل (٣ـ ١) . مع أن كل من هذه الخواص يمكن تغييره بتطبيق بيسيك المرئية ، فكثير منها تستعمل فى حالتها الافتراضية.

* لون الخلفية الأرضية الافتراضى أبيض . وهذا يعنى أن GDI يفترض أن الأرضية الخلفية لمنطقة عميل ويندو بيضاء .

* لون الفرشاة الافتراضى أبيض. أى أشكال ترسمها GDI مثل المستطيلات والمضلعات التى ترسمها ستظهر سوداء أمام الأرضية الخلفية البيضاء .

* لون النص الافتراضى أسود .GDI سيكتب رموز سوداء على حقل أبيض .

* لون القلم الافتراضى أسود. أى خطوط ومستطيلات وقطع ناقصة ومضلعات التى ترسمها GDI ستظهر سوداء أمام الخلفية الأرضية البيضاء .

* طور الأرضية الخلفية معتم ، ومزيل لأى صورة موجودة فعلاً على الشاشة . الخطوط ذات النقط والشرط ترسم بالأسود ، ولكن المسافات البيضاء بين الشرط تزيل أى رسم موجود.

* منطقة القص الافتراضية هى منطقة العميل . نشاط الرسم خارج منطقة العميل غير مرئى. كل الخطوط والمستطيلات، و bitblt (نقل قالب الرقم الثنائى : bit block transfer) والنص سيتم قصها عند أطراف منطقة العميل (Client area .

* مكان الصورة الداخلية : منطقة العميل: لويندو الأساسى تطبيق ويندوز والمتاح للاستعمال بواسطة التطبيق .	Client area
* حالة تعتبر أنها موجودة إلا إذا عرفت : بواسطة المستخدم أو المطور .	default

* أصل الافتراضى لاحداثى نظام الرسومات هو الطرف العلوى الأيسر لمنطقة العميل والموضع الحالى للقلم، وتعاد تهيئة الفرشاة لموضع O، O هذا .

إنشاء محيط / عرض : (Creating a display - Context)

لإنشاء محيط / عرض ، فنحن عادة ندعوا وظيفة( ) GDI's Get DC والمرور

لـ GDI مقبض النافذة التى نريد استعمالها لخرج الرسومات . المقبض (handle) ببساطة هو رقم من جدول داخلى تستعمله ويندوز لتحديد محيطات (بيئات) مختلفة التى قد تحفظها . يمكننا استعمال خاصية البيسيك المرئية hWnd للحصول على مقبض (النافذة) الشكل التى نريد استعماله لرسم( ) Get DC تعيد مقبض لمحيط / العرض الذى تم إنشاؤه لنا . نمرر هذا المقبض لوظائف الرسومات التى ندعوها فى GDI.

تحرير محيط / عرض :

لأن ويندوز تحفظ عدد نهائى لمحيطات العرض ، يجب أن نحرر دائماً محيط / عرض بعد أن نكون أنهينا معها . وإلا ، فان تطبيق آخر قد لا يكون قادراً على إنشاء محيط / عرض ويمكن أن يتم منعه من الكتابة للشاشة .

ونحرر محيط/ عرض بدعوة وظيفة () GDI's Release DC . تطبيقات تعامل جيداً تنشىء عادة محيط/عرض عند بداية وظيفة والتى تعطى خرج رسومات وتحرر محيط/ العرض عند نهاية الوظيفة . بعض برامج الرسوم المتحركة تنشىء محيط/ عرض عند إصدارها وتحررها عندما يتم إنهائها . هذا ذو كفاءة أكثر من إنشاء وتحرير محيط/ عرض لكل إطار فى تتابع الرسوم المتحركة .

حفظ واسترداد محيط / عرض :

يمكننا حفظ واسترداد محبط / عرض بدعوة وظيفة ( )GDI's Save DC . هذا مريح إذا كان تطبيقنا قد غير الخواص الافتراضية . وظيفة ( ) Save DC تحفظ معلومات محيط / العرض على كومة محيط (Context). ( ) Save DC تعيد مقبض للمعلومات التى اختزنتها . ولاسترداد محيط / عرض تم حفظه سابقاً ، ندعوا : ( ) Restore DC فىGDI .

( ) Save DC، ( ) Restore DC تستعملان غالباً فى تطبيقات الرسوم والتى

تعالج أكثر من نافذة عرض واحدة . حفظ واسترداد DCs بـ ( ) Save DC ، Restore DC() أكثر كفاءة من إعادة تشكيل الخواص الافتراضية فى كل مرة تحتاج أن نستعمل الـ DC .

محيطات / أداة متسقة : (Compatible device - Contexts)

محيط (أداة متسقة هو محيط / عرض محاكى . محيط / الأداة المتسقة يعود لقالب ذاكرة مشكَّلة مثل(bitmap) (نظام بايت فى ذاكرة عرض أو ذاكرة تقليدية يمثل سطح عرض واقعى يمكن عمل رسومات عليه) . bitmap عبارة عن مصفوفة مستطيلة من الأرقام الثنائية والتى يمكننا الرسم عليها وأداء عمليات رسم أخرى : وهى مفيدة خاصة لاختزان الاطارات لتتابع رسوم متحركة .

وحتى يتم نسخ محتويات (bitmap) لنافذة التطبيق على الشاشة ، فإنbitmap يجب أن تمتلك كثيراً من نفس خواص DC مثل النافذة .

محيط / الأداة المتسقة يرمز لأداة فى ذاكرة [الـ bitmap المتسق مع محيط / عرض النافذة على الشاشة] .

أدوات الرسم :

الـ GDI تقدم بياناً مفصلاً أكثر إمتداداً لأدوات الرسم عن بيسيك المرئية . وفى الحقيقة ، فإن كثيراً من مبرمجى ويندوز يصرون على أن الطريقة الوحيدة لاستغلال البيسيك المرئية هو استعمالها كنقطة مدخل داخل الخواص القوية لـ GDI.

أدوات الرسم المقدمة بواسطة gdi.exe تتضمن أقلام ، وفرش وأطقم حروف (Fonts) . الأقلام ترسم خطوط والأشكال (Shapes). الفرش تملأ المنطقة باللون . أطقم الحروف تقدم نص ، وتعليقات (Captions) وعناون (titles).

قبل أن نتمكن من استعمال أداة رسم يجب أن ننشئها وننتقيها . وعندما ننتهى من استعمالها ، يجب أن نشطبها . وظائف أدوات الرسم هى روتينات GDI

متخصصة والتى تنشيء وتنتقى وتشطب أدوات الرسم . وظائف وصف الرسم هى روتينات متخصصة تعدل الطريقة التى تعمل بها أدوات الرسم .

ونحن نستعمل أدوات رسم لأداء عمليات خرج . الأقلام ، والفرش وأطقم الحروف عبارة عن أدوات رسم . عمليات الخرج تتضمن خطوط ، ومضلعات (Polygons) ومستطيلات وقطاعات ناقصة، ونصوص وbitblts (اختصار لـbit block transfer) أى نقل قالب الرقم الثنائى [وتدعى أيضاً رسومات القوالب ومصفوفة الرسم ، و ما وراء الملفات (metafiles) وأشياء أخرى .

وظائف أدوات الرسم :

تستعمل وظائف أدوات الرسم لإنشاء وانتقاء وشطب أدوات الرسم . قبل أن نتمكن من استعمال قلم أحمر مثلاً ، يجب أن ننشؤه وننتقيه داخل محيط / العرض (display - Context) . وعندما ننتهى من استعمال القلم الأحمر ، نلغى انتقائه باستعادة القلم الافتراضى لبيئة / العرض. حينئذ ، نلغى (ندمر) القلم الأحمر ونحرر الذاكرة التى نشغلها .

إنشاء أدوات الرسم :

لإنشاء قلم جديد ، نستعمل وظيفة () GDI's Create Pen . لإنشاء فرشاة مصمته ، ندعوا () Create Solid Brush. لإنشاء فرشاة ذات موديل ، ندعوا ( ) Create Hatch Brush أو ( ) Create Pattern Brush . أى أداة رسمGDI ننشؤها تدعى أداة رسم معتادة (Custom). كذلك ، يمكننا إختيار من مجموعة من أدوات رسم داخلية (built - in) والتى تحفظها ويندوز، وتسمى أشياء الرصيد (Stock objects). فهى تتضمن أدوات مثل فرش شفافة وأقلام بشرط . وندعوا وظيفة () GDI's Get Stock Object لاستعمال واحد من أشياء الرصيد هذه .

استعمال أدوات الرسم :

لاستعمال أداة رسم أنشأناها ، يجب أن ننتقيها داخل محيط/العرض. ندعوا GDI's Select Object ( ) لعمل ذلك .

شطب أدوات الرسم :

بعد أن ننتهى من استعمال أداة رسم معتادة (Custom)، يجب أن ندمرها . تدمير أداة رسم يحرر الذاكرة التى إحتلتها . التطبيقات التى تعامل حينئذ تدير الذاكرة بعناية لأن الذاكرة هى مورد نظام مشارك . أكثر أهمية ، لا يمكننا تحرير محيط/ نص بأمان حتى نكون قد شطبنا أى أدوات رسم والتى انتقيناها فيها .

لإلغاء انتقاء أداة رسم ، ندعوا ()Select Obj وننتقى أداة رسم إفتراضيه مرة أخرى داخل محيط/ العرض. لتدمير أداة رسمCustom ، ندعوا ()Delete Object.

وظائف صفة (attribute) رسم :

وظائف الصفات المميزة للرسم هى روتيناتGDI متخصصة والتى تعدل الطريقة التى تعمل بها أدوات الرسم . هذه الوظائف يمكنها تغيير الخلفية ، وطور الرسم، ولون النص، وتصرف bitblt .

الوظائف المميزة للخلفية :

لتهيئة لون الخلفية، ندعوا وظيفة ( ) Set BK Color . وتهيئة طور الخلفية ، ندعوا وظيفة () Set BK Mode. إذا كان طور الخلفية معتم ، أى فجوات فى الخطوط ذات الشرط والملء ذات النماذج ستهيأ للون الخلفية . وضع خطوط تحت الصور سيتم تجاوزه ويزال. إذا كان طور الخلفية شفافاً، فإن GDI لن يعدل عناصر الصورة تحت هذه الفجوات عند رسم خطوط مهيأة ، وحشو نماذج وحشو ذو شرط. كذلك ، فإن طور الخلفية يؤثر أيضاً على كيفية كتابة النص . الجزء الغير مهمش لجزء Cel النص إما معتم أو شفاف ، بناء على طور الخلفية الحالى.

[Cel : صورة مطلية على أسيتات كما تستعمل فى رسوم متحركة لفيلم تقليدى ، أو المسافة المستطيلة التى يشغلها رمز واحد فى طاقم حروف محدد .

أطوار الرسم :

طور الرسم الافتراضى ، وأى رسم نرسمه سيغطى أى شىء موجـود فعلاً على سطـح العرض. يمكننا استعمال ( )GDI's Set Rop2 لتغيير طور الرسم الذى يستعمله GDI لأقـلام وأشياء مملـوءة (محشوة). أطـوار الرسـم المدعمـة تتضمن NOT،AND،XOR،OR وأخرى . المعاملات البوليه هذه تسمى شفرات تشغيل/ الهيكل الخطى (raster - operation codes) بمبرمجى ويندوز. فمثلاً ، شفرة تشغيل / الهيكل / الخطىR2 - NO T هى أوعامل مقصور- exclusive) or operator (مفيد فى علامات تجميع / مطاط (rubber - banding) .

الخواص المميزة الممتدة bitblt :

يمكن لـ GDI أن تمد أو تضغط أى bitmap (نظام من البيانات فى ذاكرة عرض أو ذاكرة تقليدية تمثل سطح عرض فعلى يمكن رسم رسومات عليه) والذى ننسخه بوظيفة bitblt. إذا كانت الصورة ()StretchBitMode لتهيئة طور الامتداد الحالى . طور الامتداد يعرف أى خطوط مسح أو عناصر صورة (pixels) سيتم شطبها عندما يتم ضغط bitmap المصدر ليناسب المكان المتاح .

Stretch Bit Mode() يؤثر على تشغيل وظيفة نسخ () Stretch Bit وليس له تأثير على وظيفة () Bit Blt والتى تنسخ a bitmap بنفس الحجم ، بدون تمدد أو ضغط .

لون النص :

اللون الافتراضى لخروج النص هو أسود . لتغيير لون النص الافتراضى ، ندعوا وظيفة GDI's Set Text Color ، مع أن قدرات النص الموجودة فى بيسيك المرئية مناسبة لمعظم التطبيقات .

الأقلام :

لإنشاء أداة رسم (قلم) ، نستعمل عادة وظيفة () GDI's Create Pen ونحدد النوع (Style) والعرض واللون الذى نريده . وتدعم ستة أنواع بواسطة ويندوز، متضمنة مصمته، وبها شرط وشفافة. العرض الافتراضى هو وحدة واحدة ، ولكن يمكننا تحديد قلم أكثر سمكاً. يمكننا دعوة () Create Pen بأى زيادة (augment) لون . هذا يعنى أننا مقيدون بـ ١٦ لون قلم إذا كنا ننفذ ويندوز على حاسب آلى مزود بـ VGA و256 لون بـ SVGA ، أو XGA أو 8514 / A .

زيادة معينة لحدود مطلوبة . زيادة :	augment

الفرش :

لإنشاء أداة رسم (فرشاة) ، فنحن عادة ندعوا وظيفة GDI's Create Solid Brush ( ) . ويمكننا طلب أى لون، وويندوز ستعطى اللون المضبوط الذى نريده أو تقريب معقول باستعمال محاكاة عناصر الصورة (dithering). إذا كنا ننفذ ويندوز على حاسب آلى مزود بـ VGA ، فإن GDI ستستعمل محاكاة عناصر الصورة إذا كان انتقاؤنا ليس واحد من الألوان الستة عشر المتوفرة على VGA . محاكاة عناصر الصورة (dithering) تعنى مزج عناصر صورة ذات درجات لون (hues) مختلفة بحيث أن النموذج الكلى يحاكى (mimics) لون مرغوب . إذا كنا ننفذ ويندوز فى طور 256 لون باستعمال موائم رسومات SVGA ، أو8514/A أو VGA ، فإنGDI سيستعمل لوحة مزج الألوان (Palette) الأجزاء الصلبة الممتدة لإنشاء درجة اللون النقية التى نريدها .

محاكاة عناصر الصورة المستعملة لتنفيذ ظل أو تلوين :	dithering

أطقم الحروف : (Fonts)

لعرض نص ، يمكننا عرض وظيفة ( ) GDI's Text Out ، والتى تستعمل الطاقم الحالى . الطاقم الافتراضى يدعى نظام (System) والذى هو Sans Serif. proportional - Spaced typeface عـدد مـن أطقم الرصيـد الأخرى متوفرة أيضاً، متضمنة Serif, sans serif, Fixed spacing & proportional spaced.لتحميل طاقم حروف ، ندعوا () Get Stock Object . وننتقى الطاقم داخل محيط/ العرض بدعوة () Select Object . يمكننا استعمال ألوان مختلفة بدعوة () Set Text Colors . لعرض النص، ندعوا () TextOut. ويمكننا تعديل الطريقة التى تعرض بها GDI أطقم بدعوة () Create Font وتحديد الحجم ، والنوع (Style) والوزن وأى خواص أخرى نريدها . ولأن مجموعة قوية لخواص النص موجودة داخل بيسيك المرئية (Visual Basic) ، فإن تطبيقات العينة فى هذا الكتاب لاتعتمد على وظيفة () GDI's Textout . وإذا كنا نخطط لاستعمال بيسيك المرئية لكتابة تطبيق كثيف الحروف (font - intensive) (مثل برنامج شكـل صفحـة ، مثلاً) فمـن الغالـب أن نستعمـل GDI بدلاً من بيسيك المرئية لمعالجة نص .

اللون : (Color)

اللون خاصية للأقلام ، والفرش ، وأطقم الحروف . ويجب أن نحدد لون عندما ننشىء قلم أو فرشاة . وأسهل طريقة لتحديد لون هى دعوة وظيفة بيسيك المرئية RGB() باستعمال الكثافات الواقعية لقذائف RGB مثل زيادات (augments). RGB() تعيد عدد طويل والذى يقابل قيمةCOLORREF المستعملة بواسطة GDI. ويمكن أن نستعملCOLORREF لتحديد لون عندما ندعوا وظائفGDI أخرى .

وصف RGB :

وظيفة ( )RGB تأخذ ثلاثة زيادات معينة (augments) والتى تمثل القاذفات الحمراء والخضراء والزرقـاء بصمام أشعة المهبـط للأجـزاء الصلبة لمونيتور العرض . وكـل زيادة معينة هو رقم فى المدى من 0 إلى 255. وقيمة 0 (صفر) تجعل القاذف مقفل (off) كلية . وقيمة 255 تجعل المونيتور عند أعلى كثافة. وتنتج (0 ,0 ,0) RGB فى أسود، لأن كل القاذفات الثلاثة مقفلة ، و(255 ,255 ,255)RGB تعطى أبيض ناصع .

إذا كان اللون الذى نريده هو لوحات مزج اللون (Pallete) الجزء الصلب لكارت الرسومات المركب ، فإن GDI يعطيه كدرجة لون نقية (pure hue) ، وإلا فإن GDI يحاكى اللون بواسطة محاكاة عناصر الصورة (dithering) .

لوحات مزج (Palettes) اللون للرسام :

لوحات مزج اللون (Palettes) عبارة عن مجموعة من الألوان . لوحة ألوان الرسام الأفتراضية (أو لوحة ألوان الرسام الجزء الصلب) تسمى لوحة ألوان رسام النظام لـ (System Palette). وهذه لوحة الرسام التى تستعملها ويندوز لعرض ومعالجة ويندوز تطبيقات ، وقوائم وصناديق الرسائل ، وصناديق الحوار والأيقونات. وويندوز تبدأ هذه اللوحة (للرسام) عند البدء ، ويمكننا دعوة وظيفة. ( ) GDI's Set System Palette Use لتغيير ألوان الجزء الصلب فى لوحة الألوان ، مع أن ويندوز ستحفظ مدخلين (أسود وأبيض) لاستعمالاتها فى ويندوز وإدارة . وأى تغييرات نؤديها للوحة ألوان النظام ستؤثر على كل الشاشة .

لوحة الألوان المنطقية :

لوحة الألوان المنطقية عبارة عن مجموعة من الألوان التى تحددها تطبيقاتنا لاستعمالاتها - شكل (٣-٢) يبين وظائفGDI التى تستعمل لإنشاء واستعمال لوحة ألوان (palette) منطقية . ويندوز تستعمل طرق مختلفة لبناء لوحة ألواننا بناء على موائم الرسومات الذى يستعمل . وإذا كنا نستعمل VGA ولوحة ألوان لون ليست واحدة من 6/ لون للأجزاء الصلبة والمتوفرة على VGA. ويندوز ستستعمل «dithering» لمحاكاة اللون . لوحة ألوان النظام الافتراضىVGA مبينة فى شكل (٣-٣).

وظائف اللون	
Local Alloc ( )	يخصص ذاكرة لحفظ هيكل لوحة ألوان (Palette)
Create Palette ()	إنشاء لوحة ألوان منطقية
Select Palette ( )	انتقاء لوحة ألوان منطقية داخل محيط/ عرض-(display / Con-text
Realize Palette ( )	تنظيم لوحة الألوان المنطقية للوحة ألوان النظام أو الأجزاء الصلبة
Palette Index ( )	كبيرة (macro) التى تنتقى لون من لوحة ألوان عند زمن التشغيل
RGB ( )	واسعة النطاق (macro) والتى تنتقى لون منطقى محاكى لعناصر الصورة دائماً ، وتنشأ عند زمن التشغيل

شكل (٣-٢) :يمكننا دعوة GDI لارسال ألوان فردية أو لوحة ألوان كلية بألوانها

(١) VGA System Palette		
color index	displayed color (٢)	RGB gun settings in hardware lookup table (٣)
0	black	0, 0, 0
1	red	127, 0, 0
2	green	0, 127, 0
3	brown	127, 127, 0
4	blue	0, 0, 127
5	magenta	127, 0, 127
6	cyan	0, 127, 127
7	light gray	191, 191, 191
8	dark gray	127, 127, 127
9	bright red	255, 0, 0
10	bright green	0, 0, 0
11	yellow	0, 0, 0
12	bright blue	0, 0, 0
13	bright magenta	0, 0, 0
14	bright cyan	0, 0, 0
15	white	0, 0, 0

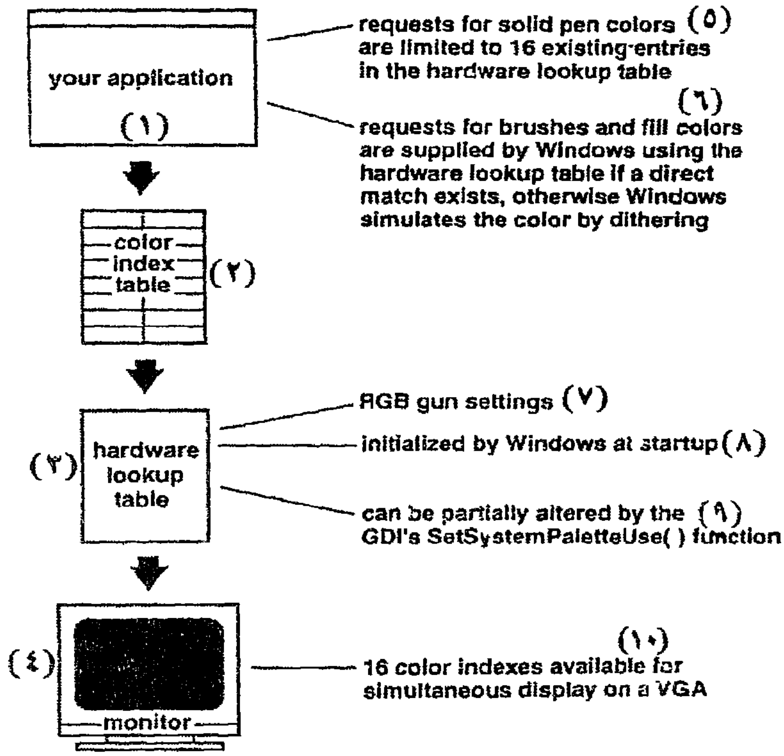
شكل (٣-٣) : ويندوز تعيد كتابة محتويات لوحة ألوان النظام عند البدء . درجات اللون المبينة هنا متوفرة لكل التطبيقات فى بيئة ويندوز

(١) لوحة ألوان VGA (٢) اللون المعروض (٣) تهيئات قاذف RGB فى جدول الرؤية أجزاء صلبة

(O) : أسود (1) : أحمر (2) : أخضر (3) : بنى (4) : أزرق (5) : أرجوانى (6) : أزرق داكن (فروزى) (7) : رمادى خفيف (8) : رمادى داكن (9) : أحمر ناصع (10) : أخضر ناصع (11) : أصفر

(12) : أزرق ناصع (13) : أحمر فوشيا ناصع (14) : فروزى ناصع (15) : أبيض

لتخصيص ذاكرة لبيانات لوحة الألوان ، ندعوا وظيفة GDI's Local Alloc() لتعريف الألوان فى لوحة الألوان، ونستعمل () Create Palette. وقبل أن نتمكن من استعمال ألوان لوحة الألوان ، يجب أن ندعوا Select Palette لتنتقى لوحة الألوان داخل محيط/ العرض . نستعمل () Realize Palette لاضافة ألوان جديدة لجدول الأجزاء الصلبة لـ RGB لـ SVGA، 8514/A ، وموائم رسومات XGA . شكل (٣ ـ ٤) يبين العلاقة بين تطبيقنا . جدول دليل اللون (لوحة الألوان المنطقية)، وجدول الأجزاء الصلبة (لوحة ألوان النظام) ومونيتور العرض . شكل (٣ ـ ٥) يوضح بدايات وعمل نماذج لعناصر الصور (dithering) التى تستعملها ويندوز إذ لم يكن اللون المطلوب متوافراً فى لوحة ألوان الأجزاء الصلبة.



شكل (٣-٤) : رسم توضيحى لمفهوم كيف تلبى ويندوز طلبات الألوان التى يعملها تطبيقنا

تابع شكل (٣-٤)

(١) التطبيق (٢) جدول دليل اللون (٣) جدول نظر الأجزاء الصلبة (٤) مونيتو جهز مراقبة)

(٥) طلب ألوان قلم صلبة محدود لستة عشرة مدخل في جدول نظر الأجزاء الصلبة.

(٦) طلبات الفرش وألوان الحشو مزودة بواسطة ويندوز باستعمال جدول نظر الأجز ـــ ـــ ة وجدت توافق ، وإلا فإن ويندوز تحاكي اللون بارتجاف .

(٧) تهيئات قاذف RGB (٨) تبدء بواسطة ويندوز عند البدء .

(٩) يمكن أن تتغير جزئياً بواسطة وظيفة : ; (GDI.s setsystem Palette Use

(١٠) دلائل 16 لون متوفرة للعرض الذى يكون في نفس اللحظة على VGA

DITHERING THRESHOLDS (٥)				
255	203	151	99	47
251	199	147	95	43
247	195	143	91	39
243	191	139	87	35
239	187	135	83	31
235	183	131	79	27
231	179	127	75	23
227	175	123	71	19
223	171	119	67	15
219	167	115	63	11
215	163	111	59	7
211	159	107	55	3
207	155	103	51	0

(١) • There are 64 dithered shades are available for each pure hue. Thresholding means that RGB (127, 0, 0), RGB (128, 0, 0), RGB (129, 0, 0), and RGB (130, 0, 0) are each displayed as normal -intensity red.

(٢) • 47 is the darkest shadow value in the lib3d 3D shading routines.

(٣) • 191 is solid light gray.

(٤) • 127 is solid dark gray and solid normal blue, green, cyan, red, magenta, and brown.

شكل (٣-٥) : إضطراب البدايات يمكن لتطبيقنا أن يستعمل غالباً أى لون لأن الـ GDI يستعمل الاضطراب لتقريب درجة لون (Hue) غير موجودة في لوحة ألوان النظام

تابع شكل (٣-٥)

(١) يوجد 64 ظلال مضطربة متوفرة لكل درجة لون نقية. البدايات تعنى أن RGB (127,0,0) RGB (128,0,0)، RGB (129,0,0)، RGB (130,0,0)، كل منها يعرض كأحمر كثافة / معتادة

(٢) 47 قيمة أكثر الظلال عتامة فى روتينات تظليل lib 3d 3D

(٣)191 هو رمادى خفيف صلب

(٤) 127 هو رمادى معتم ، وأزرق صلب معتاد ، وأخضر ، وأزرق داكن (Cygn)، وأحمر وأرجوانى(magenta) ، وبنى

(٥) بداية ترددية .

عمليات الخرج:

ويندوز GDI تعطى عديد من عمليات الخرج والتى تتضمن خط ومستطيل ومضلع وقطع ناقص ، و bitmap ومنطقة ، وقالب ذاكرة (Clipboard) وما وراء الملف (metafile) وطاقم حروف ورسم تفصيلى (mapping).

وندعوا الوظيفة () GDI's Move To لتحديد موضع القلم الحالى . ولا ترسم رسومات بواسطة () Move To . وأى خط تالى (Subsequent) نرسمه سيستعمل موضع القلم الجديد كنقطة بدايته .

وظيفة ()GDI's Line To ترسم خط من موضع القلم الحالى لنقطة نحددها عندما ندعوا ()Line To ، وموضع القلم الحالى يتم تحديثه.وأى خط لاحق سيبدأ عند نهاية الخط السابق . ويرسم الخط باستعمال اللون الحالى والنوع والسمك يمكن استعمال () Polyline لرسم مجموعة من قطع الخط الموصلة . وعندما ندعوا وظيفة ()GDI's Polyline ، فنحن نقدم مصفوفة للاحداثيات التى تمثل محاور الخطوط المتعددة (Polyline). الخطوط المتعددة ترسم باستعمال اللون الحالى، والنوع والسمك.

وظيفة ()GDI's Rectangle تستعمل لرسم مستطيل باستعمال أربعة نقط والتى نقدمها كخلاصات (arguments) . المستطيل يرسم باستعمال لون ، ونوع ، وسمك القلم الحالى ويملأ الداخل باستعمال الفرشاة الحالية.

ويمكن أن ندعوا وظيفة () GDI's Round Rect لرسم مستطيل ذو أركان دائرية باستعمال أربعة نقط والتى نقدمها كخلاصات. المستطيل مرسوم باستعمال اللون ونوع وسمك القلم الحالى . ويملأ الداخل باستعمال الفرشاة الحالية . وتحدد أشكال الأركان الدائرية بواسطة متغيرات قطع ناقص والتى نقدمها عندما ندعوا () Round Rect .

وظيفة () GDI's Polygon تنشىء مضلع متعدد الجوانب من مصفوفة من النقط التى تعطيها . إذا كانت النقط النهائية لا تغلق المضلع، فإن ()Polygon ستقفله لنا بربط النقطة النهائية مع النقطة الأولى. المضلع مرسوم باستعمال اللون ونوع وسمك القلم الحالى . ويملأ الداخل بالفرشاة الحالية باستعمال سجل حشو (Filling algorithm) والذى نحدده بدعوة () Set Poly Fill Mode . لمعظم المضلعات البسيطة، فإن طور الحشو الافتراضى كافى .

> **رسومات ثلاثة أبعاد بالحاسب الآلى** — **bounding box**
>
> صندوق ذو ستة أوجه أو رسومات حاسب آلى ذات بعدين، مستطيل يحيط محاور شىء . يسمى Stand - in كذلك .

ويمكننا رسم دوائر وقطاعات ناقصة بدعوة () Ellipse فى الـ GDI. نقدم أربعة مجموعات لاحداثيات X-Y لتحديد صندوق "bounding box" والذى يحيط بالدائرة أو القطع الناقص . إذا كان الصندوق المحيط مربع ، فإن GDI ترسم دائرة. إذا كان الصندوق المحيط مستطيل، فإن GDI ترسم قطع ناقص . ويرسم الشكل باستعمال اللون، والنوع والسمك للقلم الحالى . ويحشى الداخل باستعمال الفرشاة الحالية .

:bitmaps

وهو عبارة عن مصفوفة من الأرقام الثنائية فى ذاكرة والتى تمثل صورة . ويمكننا

الرسم على bitmap تماماً مثل ما نرسم على الشاشة ويمكننا نسخ bitmaps من ذاكرة الشاشة ، ومن الشاشة للذاكرة ومن ذاكرة لذاكرة . ومتتخبات من وظائفGDI bitmap المفيدة مبنية فى شكل (٣ـ٦) .

وظائف Bitmap	
إنشاء محيط / عرض ذاكرة لـ bitmap المختفى	Create Compatible DC ( )
إنشاء bitmap متسقة مع محيط / العرض (display / context)	Create Compatible Bitmap ()
إنتقاء bitmap داخل «محيط / عرض» أو داخل «محيط/ عرض» ـ ذاكرة	Select Object ( )
نسخ bitmap من محيط عرض واحد (أو أداة)لآخر.	Bit Blt ( )
تدمير bitmap	Delete Object ( )
ينسخ ويعيد تحجيم bitmap bitmap(resizes) (resizes) ليملأ محيط / عرض الهدف أو محيط الأداة .	Stretch Bit ( )
استرجاع معلومة (info) من رأس bitmap بخصوص العرض ، والعمق ، إلخ	Get Object ( )

شكل (٦ـ٢) : الـ GDI تعطى تسيق (assortment) من الوظائف لانشاء وانتقاء ونسخ ومراقبة

إنشاء bitmap :

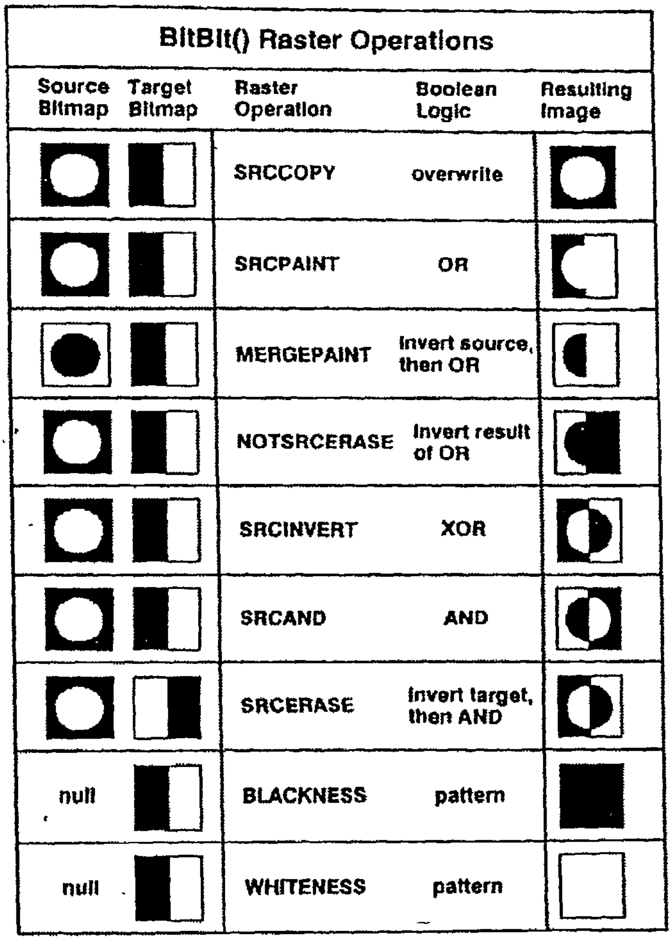
لانشاء "a bitmap" ، يجب أولاً أن ندعوا الـ GDI لانشاء محيط / أداة متسق فى ذاكرة هذا المحيط/ للأداة المتسق يمتلك خواص تشبه محيط/ العرض للشاشة .

ونحن ننشيء «محيط / أداة» متسق بدعوة Create Compatible DC() . المحيط (البيئة) المنشأ حديثاً يدعى أيضا «محيط / عرض» الذاكرة . ونحن غالباً نستعمل محيط عرض / أداة لانشاء bitmap والمتسق مع كلا محيط «عرض / الذاكرة»، «ومحيط / العرض» لويندو التطبيق على الشاشة . وندعوا Create Compatible( ) Bitmap لانشاء الـ bitmap. وقبل أن نبدأ الرسم على bitmap يجب أن ننتقيها فى محيط عرض / الذاكرة بدعوة Select Object ( ) . ويمكننا إنشاء رسومات علىbitmap بنفس الطريقة التى نرسم بها على ويندو التطبيق . ونستعمل المقبض (handle) لمحيط عرض الذاكرة (بدلاً من محيط / العرض) عندما ندعوا الـ GDI. طريقة أخرى لانشاء الرسومات علىbitmap هى أن نرسم على ونيدو التطبيق الأساسى الخاص بنا وننسخ الصورة المنتهية لـ bitmap . النسخ فى الاتجاه العكسى من الـ bitmap للويندو عبارة عن جزء هام لصور ويندوز المتحركة (animation).

تحريك a bitmap :

يمكننا نسخ a bitmap من موقع لآخر بدعوة وظيفة GDI's Bitblt ( ) . يمكننا استعمال عمليات هيكل خطى مختلفة لاعطاء نتائج مختلفة ، كما هو مبين فى شكل (٣ ـ ٧) . وظيفة Bit Blt ( ) مفيدة لرسوم الاطار المتحركة ولتنفيذ وظائف إيطال(undo) والانعاش(refresh) بتحريك bitmap صفحة مختفية لويندو المرئية.

اذا كان من الواجب مد bitmap أو ضغطها حتى تناسب جهة الوصول ، يمكننا استعمال وظيفة GDI'S Stretch Blt ( ) . وكلا BitBlt () stretch Blt ( ) يمكنها استعمال عمليات الهيكل الخطى (raster) للتحكم فى النتيجة عندما توضع bitmap على الصورة الموجودة . ويمكن أن نحرك جزء مستطيل لـbitmap المصدر إلىbitmap الهدف . هذا هو كيفية تطبيق عينه يمكنه تنفيذ خلفية دوارة . فهوينسخ مستطيلات منتقاة من مختفية والتى أكبرمن الويندوعلى الشاشة .



BitBlt() Raster Operations				
Source Bitmap	Target Bitmap	Raster Operation	Boolean Logic	Resulting Image
		SRCCOPY	overwrite	
		SRCPAINT	OR	
		MERGEPAINT	Invert source, then OR	
		NOTSRCERASE	Invert result of OR	
		SRCINVERT	XOR	
		SRCAND	AND	
		SRCERASE	Invert target, then AND	
null		BLACKNESS	pattern	
null		WHITENESS	pattern	

شكل (٧-٣) : بعض عمليات الهيكل الخطى الكثيرة المتوفرة مع وظيفة
GDI,s Bitblt ( )

شطب bitmap :

يمكننا شطب a bitmap بعد أن ننتهى من استعمالها بدعوة وظيفة GDI's De-lete Object ( ) وهذا يخلى الذاكرة المستعملة بواسطة bitmap.

المناطق : (Regions)

أصناف وظائف المنطقة تزود بواسطة GDI . فهى مناطق ملء (Fill) ومناطق

قص (Clipping). وظائف المل وتنشء مناطق بشكل المضلعات والتى يمكن ملئها بفرشاة. مناطق القص عبارة عن منافذ العرض (View Ports) والتى تستعمل لقص خرج الرسومات .

مجموعة ثانوية لشاشة العرض: منفذ العرض :	Viewport

● انشاء منطقة ملء :

يمكنا انشاء منطقة ملء مستطيلة بدعوة ( ) Create Rect Rgn . لانشاء «منطقة / ملء» فى شكل مستطيل دائرى الاركان ندعوا: لانشاء قطع ناقص كمنطقة مملوءة ندعوا ( ) Create Round Rect Rgn. لانشاء قطع ناقص كمنطقة مملوءة ندعوا ( ) Create Elliptic Regn

ملء منطقة :

يمكننا استعمال ( ) Fill Rgn لملء منطقة / مملوءة (Fill - region) بفرشاة والتى نمررها كخلاصة (argument) . الفرشاة لا تحتاج أن تكون فرشاة حالية: تستعمل ( ) Paint Rgn لملء منطقة / مملوءة بالفرشاة الحالية modelling عمل موديل ومكتبة الظل تستعمل وظيفة ( ) GDI's Fill Rgn لملء الأسطح المضلعة (Facets) لأشياء صلبه .

إنشاء شكل هندسى يمثل شىء ذو ثلاثة أبعاد :	modeling

استعمال مناطق القص :

لانشاء منطقة قص ، نْدعوا أولاً ( ) Create Rect Rgn لانشاء منطقة . حينئذ نستعمل ( ) Select Clip Rgn لأنتقائها كمنطقة قص حالية . وأى رسومات تالية كخرج سيتم قصها عند أطراف المنطقة .

سطح مضلع ليستعمل لإنشاء :	Facet
موديلات3D مصمته تنشأ بطريقة B - rep	

أطقم الحروف والنص :

وظائف أطقم الحروف المزودة بواسطة GDI تندرج تحت نوعين : فهى وظائف حروف ووظائف نص . وظائف أطقم الحروف تنشىء وتنتقى أطقم حروف . طاقم الحروف هو نوع من الرموز الهجائية العددية .وظائف النص تعرض الرموز على الشاشة. إصدار صغير (Retail) لويندوز يعطى عدداً من أطقم الحروف الموجودة بها والتى يمكن لتطبيقنا أن يستعمله. أطقم الحروف المفيدة ووظائف النص التى تزودها GDI مبينة فى شكل (٣ـ٨) .

وظائف Bitmap	
Text Out ()	تعرض صف نص باستعمال طاقم الحروف الحالى
Get Stock Object ()	يسترجع مقبض لطاقم حروف النظام
Select Object ( )	تنتقى طاقم حروف داخل «محيط/ عرض» أومحيط أداة
Set Text Color	تهىء لون النص (الأمامى : «Foreground» فقط).
Set BK Color	تهىء لون الخلفية Cel النص
Set BK Mode ( )	تهىء خلفية النصCel إلى معتم أو شفاف
Create Font ( )	تنشىء طاقم حروف منطقى بارتفاع محدد، وكذلك الوزن والنوع والخ .

شكل (٨ـ٢) : الـ GDI تعطى وظائف لانتقاء أطقم حروف بأنواع، وأحجام وألوان مختلفة

الرسم التفصيلى (Mapping) :

الـ GDI، تعطى مجموعة من وظائف وإحداثيات (Coordinates) والتى ترسم (Map) رسومات من نظام إحداثى (Coordinate) واحد لآخر . فى طور الرسم الافتراضى هذا، فإن GDI يرسم بالتفصيل (map) إحداثيات X - Y على أسس فردية لعناصر الصورة فى ويندو التطبيق. الأصل، المقدم بالاحداثيات(0, 0)، يوجد عند الركن الأيسر العلوى لسطح الرسم . نظام الاحداثى هذا يسمى MM . TEXT

نظم إحداثيات أخرى هى تطبيقات رسوم متخصصة مفيدة ، مثل إعداد ، تخطيط الصفحة (Pagelayout) وهكذا.طور MM-LONGLISH يرسم كل من إحداثيات Y - X إلى وحدات 0.01 بوصة على سطح العرض طور -TheMM LOMETRIC يرسم إحداثياتY - X إلى وحدات 0.1 ميللمتر على سطح العرض .

استعمال منفذ الرؤيه وتدريج ويندوز (Scaling) :

أطوار الرسم التفصيلى (mapping) تستعمل منافذ رؤية (Viewports) وتدريج ويندوز . عند الكلام عن الرسم التفصيلى فإن ويندوز لاتعنى ويندو تطبيق، ولكن مستطيل تدريج نظرى (abstract Scaling) . ويعرف منفذ الرؤية باحداثيات طبيعية باشغال مكان إحداثيات طبيعية . من الناحية الأخرى تدريج ويندوز (Scaling Window) يعرف باحداثيات منطقية ليس لها علاقة بعناصر الشاشة . وتدريج ويندوز (Scaling) هذا يقال أنه يشغل مكان إحداثى منطقى. عندما يستدعى وظيفة رسومات GDI ، فإننا نعطى إحداثيات منطقية كخلاصات (arguments) . طور رسم (mapping) الحالى يحدد كيف يقوم برسم تلك الاحداثيات المنطقية للشاشة (والتى تستعمل إحداثيات طبيعية) أو لمنفذ الرؤية، هو مستطيل قص على الشاشة بالافتراض فهو مشروع رسم (mapping) 1 :1 .

الاحداثيات المنطقية التى نعطيها تم رسمها مباشرة لمكان إحداثى / طبيعى . ويندو التطبيق الأساسى فى مكان إحداثى / طبيعى) .

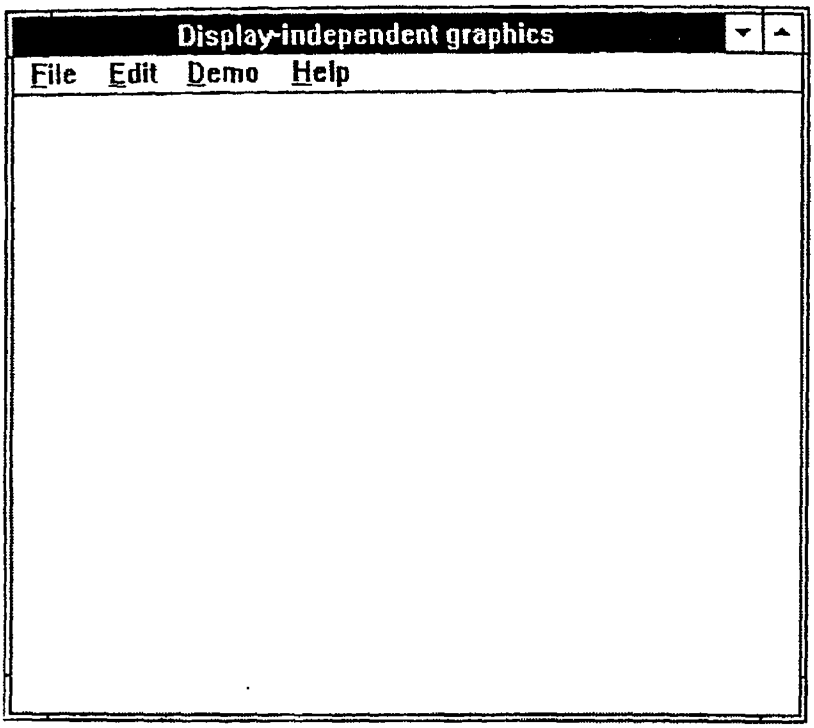
تطبيق عينه :

صورة الشاشة فى شكل (٣ ـ ٩) تبين عرض البدء من .detect.exe تطبيق العينة . هذا البرنامج يدرب كثيراً من وظائف رسومات GDI ، ويعطى مثال عامل لرسومات متواصلة فى بيئة متعددة العمل ومتعاونة . تطبيق العينة سينفذ فى أى طور عرض ويندوز، والصورة ستنعش جيداً . ويمكن أن نغطى ويندو .detect.exe يويندو تطبيق آخر أو بمجموعة برنامج ؛ وعندما تكتشف الويندو أن رسوماتها لن تتأثر . ويمكننا تحريك ويندو.detect.exe خارج طرف الشاشة ، وعندما نحرك الويندو مرة أخرى على الشاشة ، فإن رسوماتها ستعرض بطريقة صحيحة. التطبيق البسيط سيكشف ويبلغ عن دقة الشاشة الحالية أيضاً ، وأقصى عدد لألوان الأجزاء الصلبة المتوفرة وطور الذاكرة الحالى الذى تستعمله ويندوز (عياريه أو مدعمة). detect exe. demo مكون من ثلاث وحدات ، متضمنة وحدة شاملة detect. bas،ووحدة شكل a detect. Frm، ووحدة بدء detect 01. bas. كل ملفات الانتاج هذه مقدمة كشفرة مصدرة مستعدة ـ لـ الاستعمال (ready - to - use) فى قائمة البرنامج . كذلك فإن ملفات المصدر موجودة على القرص المصاحب ، مع ملف مشروع detect.prj .

ماذا يفعل البرنامج ؟

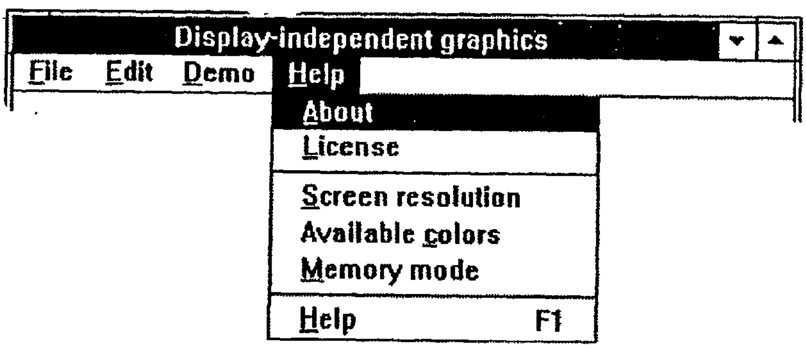
دليل المستعمل :

دليل المستعمل : يمكن استعمال إما فأرة أو لوحة المفاتيح للتدريب مع تطبيق العينة . كذلك ، فإن القوائم تدعم مفاتيح ذاكرة (mnemonic) .

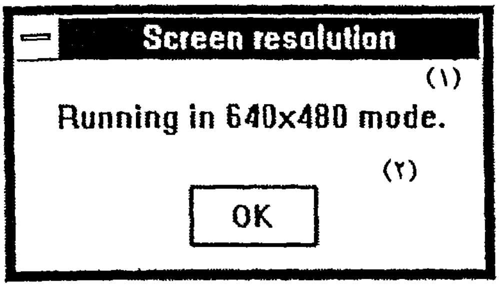


شكل (٣-٩) : نافذة العرض من تطبيق العينة. البرنامج التوضيح يعطي تخيل (imagery) رسومات مستقلة عن العرض خلال أطوار الشاشة

استقلال الأداة :



شكل (٣-١٠) : قائمة Help من تطبيق العينة تعطى معلومات عن حالات زمن التشغيل مثل دقة الشاشة، وأكبر عدد من الألوان المعروضة، وطور الذاكرة الذى يتم استعماله بواسطة ويندوز (حقيقى أو عيارى أو مدعم)



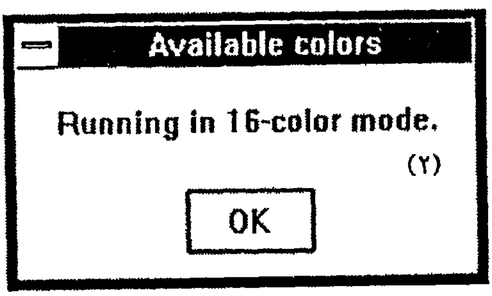
شكل (٣-١١) : صندوق رسالة دقة الشاشة من تطبيق العينة

(١) دقة الشاشة

(٢) يعمل فى طور 640 × 486

كما نرى فى شكل (٣ ـ ١٠) ، يمكننا إستعمال قائمة Help للحصول على معلومات عن دقة الشاشة ، وأقصى عدد لألوان الأجزاء الصلبة المتاحة للاستعمال ، وطور الذاكرة الحالى المستعمل بواسطة ويندوز. صورة الشاشة فى شكل (٣ ـ ١١) تبين صندوق رسالة معتاد والذى يعرض عندما ننتقى دقة شاشة (Screen res-olution) من قائمة Help . الصورة المبينة فى شكل (٣ ـ ١٢) هى تقرير معتاد والذى يعرض عندما ننتقى ألوان متوفرة (Available Colors) من قائمة Help. صورة الشاشة فى شكل (٣ ـ ١٣) توضح صندوق رسالة معتاد والذى يظهر عندما ننتقى طور ذاكرة (Memory mode) من قائمة Help.

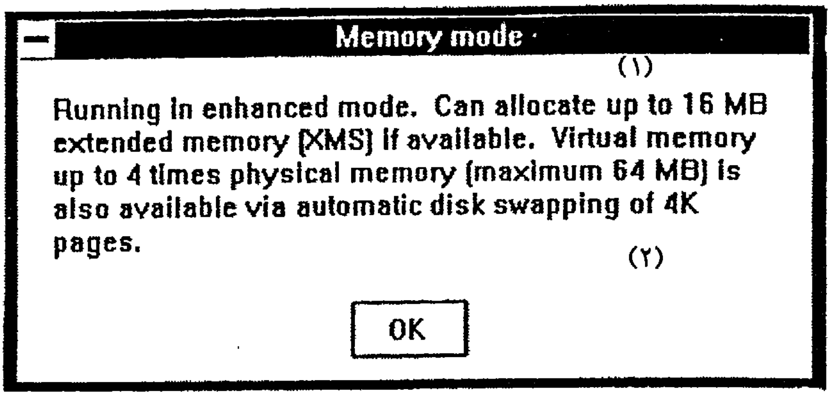
(١)



شكل (٣-١٢) : صندوق رسالةالألوان المتوفرة من تطبيق العينة

(١)الألوان المتوفرة

(٢) يعمل فى طور 16



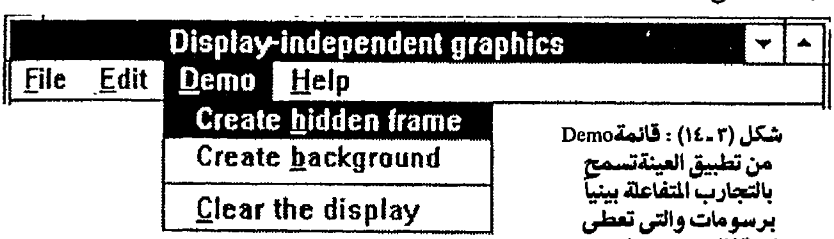
شكل (٣-١٣) : صندوق رسالة طور الذاكرة من تطبيق العينة

(١) طور الذاكرة

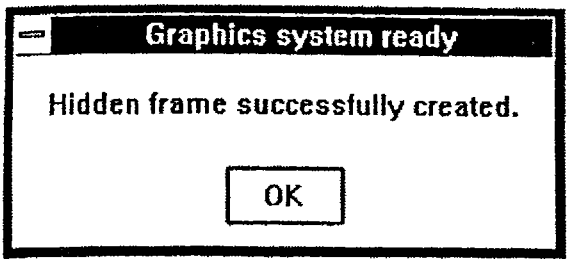
(٢) يعمل في طور مدعم . يمكن أن يخصص حتى16 MB كذاكرة ممتدة [XMS] إذا توفرت الذاكرة الواقعية حتى أربعة مرات للذاكرة الطبيعية [64MB أقصى] متوفرة أيضاً خلال مبادلة قرص أوتوماتيكية لصفحات 4K

الرسومات المتواصلة : (Persistent graphics)

يمكننا انتقاء بنود من قائمة Demo للتدريب مع الرسومات المتواصلة كما في شكل (٣ _ ١٤) . لإنشاء صورة ، أولاً ننتقي اطار Create Hidden من القائمة العامة (Demo). شكل (٣ _ ١٥) يبين صندوق الرسالة الذى يظهر بعدbitmap مختفى قد تم إنشاؤه . bitmap هذه ستستعمل لإنشاء نسخة مزودجة لمنطقة عمل ويندوز . وسيتم نسخ bitmap للويندوزعندما تحتاج الصورة لانعاش ولا تقص أو تتحرك أو لا تغطى . لانشاء صورة ، ننتقى create background من قائمة عامة (Demo). ويتم رسم مجموعة من الظلال الرقيقة (Subtle) ومنحدرة (gradients) كما في شكل (٣_١٦) .



شكل (٣-١٤) : قائمةDemo من تطبيق العينةتسمح بالتجارب المتفاعلة بينيا برسومات والتى تعطى استقلال / عرض إسمى (Nominal)



شكل (٣ـ١٥) : صندوق رسالة يعطى تقرير حالة لـbitmap المختفية والمستعملة لاختزان الرسومات المتواصلة عند زمن التنفيذ . الصورة فى bitmap المختفية تكتب لويندو العرض عندما لا يغطى الويندو أو لا يقص ويحتاج أن ينعش .

التطبيق أنيق بدرجة تكفى لتجاهل الأخطاء البسيطة . وإذا حاولنا إنشاء تطبيق إطار مختفى بعد أن تم إنشاء واحد ، فإن صندوق الرسالة المبين فى الجزء العلوى فى شكل (٣ ـ ١٧) سيتم عرضه . وإذا حاولنال إنشاء صورة قبل أن ننشىء bitmap مختفية ، فإن صندوق الرسالة المبين فى الجزء المنخفض لشكل (٣ ـ ١٧) سيتم عرضه .

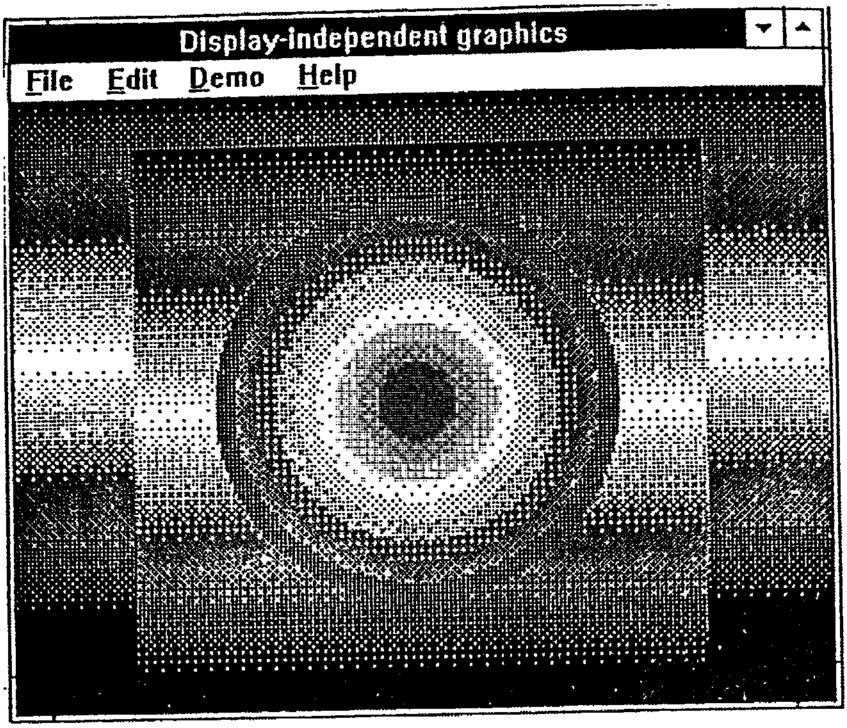
كيف تعمل شفرة المصدر :

برنامج شرح .detect. exe يعطى برنامج platform قوى لتطوير الرسومات المتحركة باستعمال البيسيك المرئية. الرسوم المتواصلة تجعل من الممكن تنفيذ تتابعات رسوم متحركة أثناء تنفيذ تطبيقات أخرى . قدرة demo لكشف دقة الشاشة وطور ذاكرة زمن التشغيل أنيق بدرجة تكفى أن نقرر إذا كنا ننفذ رسوم متحركة من قرص أو من ذاكرة. تطبيق عينة detect.exe يعطى أساساً قوياً والذى تم بناء كل صور demos المتحركة فى الكتاب.

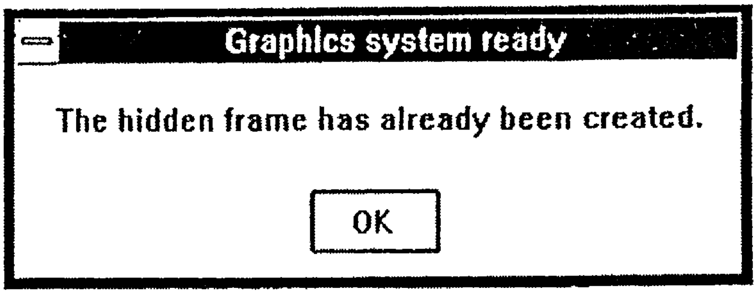
كيف تعمل الوحدة الشاملة :

قائمة الوحدة الشاملة detect.bas مبينة فى شكل (٣ ـ ١٨) . هذه الوحدة تحتوى على إعلانات (declarations) لثوابت ومتغيرات وتلك التى يمكن استعمالها بواسطة كل الوحدات فى المشروع . كذلك، فإن الوحدة الشاملة تحتوى على إعلانات (declarations) الثوابت والتى ستستعمل عند دعوة وظائف فى مشترك أداة الرسومات ويندوز GDI . كذلك، تعطى على إعلانات لوظائف GDI

والتى تجعل من الممكن لشفرة بيسيك المرئية لاستعمال روتيناتGDI لأداء عمليات رسومات فى وقت التنفيذ .



شكل (٣-١٦) : صورة لتحاكى عناصر الصورة (dithered) منشأة بواسطة تطبيق العينة . الصورة ستتواصل وتنعش بطريقة صحيحة إذا لم تقص نافذة العرض بواسطة طرف الشاشة ، أولا تغطى بحركة ويندو أخرى .



شكل (٣-١٧) : صناديق الرسائل تستعمل بواسطة تطبيق العينة لتنصحنا لاستثناءات زمن التشغيل (أخطاء) والتى قد تم شطبها .

المتغيرات العامة :

مجموعة من الثوابت العامة والمتغيرات العامة معلنة عند السطور 0036 إلى 0042. الثوابت والمتغييرات التى تصف الشكل الأساسى معلنة من السطر 0047 إلى السطر 0054 . المتغيرات التى تستعمل لعدد من أغراض التدبير المنزلى العامة ، معلنة فى السطور 0059 إلى 0068. كل هذه الثوابت والمتغيرات تم التعليق عليها فى شفرة المصدر، لذلك يمكن أن نفهم الغرض المخصصة له .

عمليات الإطار المختفى :

الشفرة عند السطور 0073 إلى 0076 تعلن مجموعة من المتغيرات التى تدعم استعمال زمن تشغيل الاطار المختفى. المتغيرات عند السطور 0073 إلى 0075 تستعمل كمقابض لمحيطات/ العرض bitmaps . المتغير عند السطر 0076 هو علم الوضع (status flag) .

الثوابت لويندوز API :

الشفرة عند السطور 0081 إلى 0103 تعلن مجموعة من الثوابت والتى ستستعمل عند دعوة وظائف فى GDI . كثير من هذه الثوابت تم التعليق عليها لتبين الغرض المخصصة له.

إعلانات وظيفة GDI :

وظائف GDI المدعوة (Collable) معلنة عند السطور 0108 إلى 0170 . هذه الوظائف مصنفة فى أنواع كما هو مبين بالتعليقات . ويلاحظ أن المجموعة النهائية لاعلانات الوظائف عند السطور 0167 إلى 0170 فى الحقيقة ترجع لزمن التنفيذ الموجود فى exe . user و Kernel . exe DLLS لمشترك برمجة تطبيق ويندوز (API : application programming interface) .

شكل (٣ـ١٨) :

```
'-----------------------------------------------------------------------
'  Reusable global module for Visual Basic graphics applications     (١)
'               that call Windows API functions.
'-----------------------------------------------------------------------
'  Source file:  DETECT.BAS
'  Release version:  1.00                      Programmer:  Lee Adams
'  Type:  Visual Basic global module for Windows applications.
'  Compiler:  Microsoft Visual Basic 1.00
'  Dependencies:  DETECT.FRM primary module
'                 DETECT01.BAS module containing Main()
'  Output and features:  Provides declarations for Windows API
'    (Application Programming Interface) functions callable by
'    Visual Basic applications at runtime, including routines
'    from Windows' GDI, USER, and KERNEL DLLs (dynamic link
'    libraries).  Also provides declarations of various variables
'    and constants.  Functions, variables, and constants declared
'    in this global module are visible throughout the project.
'  Publication:  Contains material from Windcrest/McGraw-Hill
'    book 4224 published by TAB BOOKS Div. of McGraw-Hill Inc.
'  License:  As purchaser of the book you are granted a
'    royalty-free license to distribute executable files
'    generated using this code provided that you accept the
'    conditions of the License Agreement and Limited Warranty
'    described in the book and on the companion disk.  Government
'    users:  This software and documentation are subject to
'    restrictions set forth in The Rights in Technical Data and
'    Computer Software clause at 252.227-7013 and elsewhere.
'-----------------------------------------------------------------------
'      (c) Copyright 1992 Lee Adams.  All rights reserved.           (٢)
'           Lee Adams(tm) is a trademark of Lee Adams.
'-----------------------------------------------------------------------

'-----------------------------------------------------------------------
'                 General constants and variables                    (٣)
'-----------------------------------------------------------------------
Global Const True = -1                            'boolean constant
Global Const False = 0                            'boolean constant
Global Const MB_OK = 0                     'OK button for message box
Global Const PIXELS = 3                          'pixel coordinates
Global Const Null = 0                                    'undefined
Global StartUp As Integer                   'tracks the startup code
Global IgnoreRefresh As Integer             'tracks refresh activity

'-----------------------------------------------------------------------
'                       Window specifications                        (٤)
'-----------------------------------------------------------------------
Global Const zWINDOW_WIDTH = 408                     'width of window
Global Const zWINDOW_HEIGHT = 346                   'height of window
Global Const zFRAMEWIDE = 400                    'width of client area
Global Const zFRAMEHIGH = 300                   'height of client area
Global HorizTwipsPixel As Single     'twips-per-pixel (horizontal)
Global VertTwipsPixel As Single        'twips-per-pixel (vertical)
Global Window_Width As Single               'runtime width of window
Global Window_Height As Single             'runtime height of window

'-----------------------------------------------------------------------
'                        Runtime conditions                          (٥)
'-----------------------------------------------------------------------
Global hDesktopWnd As Integer                     'handle to desktop
Global hDCcaps As Integer                'display-context for desktop
```

```
Global DisplayWidth As Integer        'horizontal screen resolution
Global DisplayHeight As Integer         'vertical screen resolution
Global DisplayBits As Integer             'number of bits-per-pixel
Global DisplayPlanes As Integer                'number of bitplanes
Global MemoryMode As Long                      'runtime memory mode
Global RetVal As Integer               'will receive GDI return value
Global RetLong As Long                 'will receive GDI return value
Global MousePresent As Integer                        'mouse active?

'------------------------------------------------------------------
'                   Hidden frame operations                    (٦)
'------------------------------------------------------------------
Global hFrameDC As Integer      'display-context for hidden-frame
Global hFrame As Integer           'handle to hidden-frame bitmap
Global hPrevFrame As Integer         'default bitmap for hFrameDC
Global FrameReady As Integer                'hidden-frame created?

'------------------------------------------------------------------
'             Constants for Windows API functions              (٧)
'------------------------------------------------------------------
Global Const SRCCOPY = &HCC0020                       'for bitblts...
Global Const SRCINVERT = &H660046
Global Const SRCPAINT = &HEE0086
Global Const WHITENESS = &HFF0062
Global Const BLACKNESS = &H42&
Global Const ALTERNATE = 1                            'for filling...
Global Const WINDING = 2
Global Const R2_COPYPEN = 13                         'for pen mode...
Global Const R2_XORPEN = 7
Global Const TRANSPARENT = 1                  'for background mode...
Global Const OPAQUE = 2
Global Const PS_SOLID = 0                              'for solid pen
Global Const PS_NULL = 5                         'for transparent pen
Global Const BLACKONWHITE = 1                   'for bitblt scaling...
Global Const WHITEONBLACK = 2
Global Const COLORONCOLOR = 3
Global Const HORZRES = 8                  'args for GetDeviceCaps()...
Global Const VERTRES = 10
Global Const BITSPIXEL = 12
Global Const PLANES = 14
Global Const SM_MOUSEPRESENT = 19         'for GetSystemMetrics()
Global Const WF_ENHANCED = &H20              'for GetWinFlags()...
Global Const WF_STANDARD = &H10

'------------------------------------------------------------------
'              GDI functions for display-contexts              (٧)
'------------------------------------------------------------------
Declare Function GetDC Lib "USER" (ByVal hWnd As Integer) As
     Integer
Declare Function ReleaseDC Lib "USER" (ByVal hWnd As Integer,
     ByVal hDC As Integer) As Integer

'------------------------------------------------------------------
'                 GDI functions for the desktop                (٩)
'------------------------------------------------------------------
Declare Function GetDesktopWindow Lib "USER" () As Integer

'------------------------------------------------------------------
'           GDI functions for creating drawing objects        (١٠)
'------------------------------------------------------------------
```

```
Declare Function CreatePen Lib "GDI" (ByVal PenStyle As Integer,
     ByVal Wd As Integer, ByVal Color As Long) As Integer
Declare Function CreateSolidBrush Lib "GDI" (ByVal Color As Long)
     As Integer

'------------------------------------------------------------------
'             GDI functions for selecting objects                (١١)
'------------------------------------------------------------------
Declare Function SelectObject Lib "GDI" (ByVal hDC As Integer,
     ByVal hObject As Integer) As Integer
Declare Function DeleteObject Lib "GDI" (ByVal hObject As Integer)
     As Integer

'------------------------------------------------------------------
'             GDI functions for bitmaps and bitblts              (١٢)
'------------------------------------------------------------------
Declare Function CreateCompatibleDC Lib "GDI" (ByVal hDC As
     Integer) As Integer
Declare Function CreateCompatibleBitmap Lib "GDI" (ByVal hDC As
     Integer, ByVal Wd As Integer, ByVal Ht As Integer) As Integer
Declare Function PatBlt Lib "GDI" (ByVal hDC As Integer, ByVal X
     As Integer, ByVal Y As Integer, ByVal Wd As Integer, ByVal Ht
     As Integer, ByVal RasOp As Long) As Integer
Declare Function BitBlt Lib "GDI" (ByVal hDestDC As Integer, ByVal
     DestX As Integer, ByVal DestY As Integer, ByVal Wd As Integer,
     ByVal Ht As Integer, ByVal hSrcDC As Integer, ByVal SrcX As
     Integer, ByVal SrcY As Integer, ByVal RastOp As Long) As
     Integer
Declare Function DeleteDC Lib "GDI" (ByVal hDC As Integer) As
     Integer

'------------------------------------------------------------------
'           GDI functions for drawing mode operations            (١٣)
'------------------------------------------------------------------
Declare Function SetROP2 Lib "GDI" (ByVal hDC As Integer, ByVal
     RasMode As Integer) As Integer
Declare Function SetBkColor Lib "GDI" (ByVal hDC As Integer, ByVal
     Color As Long) As Long
Declare Function SetBkMode Lib "GDI" (ByVal hDC As Integer, ByVal
     BkMode As Integer) As Integer
Declare Function SetPolyFillMode Lib "GDI" (ByVal hDC As Integer,
     ByVal PolyMode As Integer) As Integer
Declare Function SetStretchBltMode Lib "GDI" (ByVal hDC As
     Integer, ByVal StretchMode As Integer) As Integer

'------------------------------------------------------------------
'             GDI functions for drawing operations               (١٤)
'------------------------------------------------------------------
Declare Function MoveTo Lib "GDI" (ByVal hDC As Integer, ByVal X
     As Integer, ByVal Y As Integer) As Long
Declare Function LineTo Lib "GDI" (ByVal hDC As Integer, ByVal X
     As Integer, ByVal Y As Integer) As Integer
Declare Function Polygon Lib "GDI" (ByVal hDC As Integer, FirstPt
     As Integer, ByVal Count As Integer) As Integer
Declare Function Rectangle Lib "GDI" (ByVal hDC As Integer, ByVal
     X1 As Integer, ByVal Y1 As Integer, ByVal X2 As Integer, ByVal
     Y2 As Integer) As Integer
Declare Function Ellipse Lib "GDI" (ByVal hDC As Integer, ByVal X1
     As Integer, ByVal Y1 As Integer, ByVal X2 As Integer, ByVal Y2
     As Integer) As Integer
```

```
Declare Function FloodFill Lib "GDI" (ByVal hDC As Integer, ByVal
     X As Integer, ByVal Y As Integer, ByVal Color As Long) As
     Integer
Declare Function SetPixel Lib "GDI" (ByVal hDC As Integer, ByVal X
     As Integer, ByVal Y As Integer, ByVal Color As Long) As Long
Declare Function GetPixel Lib "GDI" (ByVal hDC As Integer, ByVal X
     As Integer, ByVal Y As Integer) As Long
'-----------------------------------------------------------------
'                  GDI functions for regions                      (١٥)
'-----------------------------------------------------------------
Declare Function PaintRgn Lib "GDI" (ByVal hDC As Integer, ByVal
     hRgn As Integer) As Integer
Declare Function CreatePolygonRgn Lib "GDI" (FirstPt As Integer,
     ByVal Count As Integer, ByVal PolyFillMode As Integer) As
     Integer
'-----------------------------------------------------------------
'  GDI, USER, KERNEL functions for various runtime conditions     (١٦)
'-----------------------------------------------------------------
Declare Function GetDeviceCaps Lib "GDI" (ByVal hDC As Integer,
     ByVal Item As Integer) As Integer
Declare Function GetWinFlags Lib "KERNEL" () As Long
Declare Function GetSystemMetrics Lib "USER" (ByVal Item As
     Integer) As Integer
Declare Function GlobalCompact Lib "KERNEL" (ByVal NumBytes As
     Long) As Long
'-----------------------------------------------------------------
'                     End of global module.                       (١٧)
'-----------------------------------------------------------------
```

بقية شكل (٣-١٨) : قائمة الوحدة الشاملة لتطبيق العينة

(١) وحدة شاملة معاد استعمالها لتطبيقات رسومات البيسيك الأساسية والتى تدعو وظائف ويندوز API.

(٢) حق النسخ Lee Addams 1992 . كل الحقوق محفوظة.Lee Adams (tm). هى ماركة مسجلة لـ Lee Adams.

(٣) الثوابت والمتغيرات العامة

(٤) مواصفات ويندوز.

(٥) حالات زمن التنفيذ

(٦) عمليات الإطار المختفي.

(٧) ثوابت لوظائف ويندوز API

(٨) وظائفGDI لمحيطات /عرض .

(٩) وظائفGDI لسطح المكتب

(١٠) وظائفGDI لانشاء أشياء رسم

تابع شكل (٣-١٨)

(١١) وظائف GDI لانتقاء أشياء

(١٢) وظائف GDI لـ bitmaps ، bitblt

(١٣) وظائف GDI لعمليات طور الرسم

(١٤) وظائف GDI لعمليات الرسم

(١٥) وظائف GDI للمناطق

(١٦) وظائف GDI، USER ،KERNEL لحالات زمن تشغيل متعددة

(١٧) نهاية الوحدة الشاملة .

شكل (٣-١٩) :

```
'-----------------------------------------------------------------
'     Reusable template for Visual Basic Windows applications
'      that use a hidden frame, display-independent graphics,      (١)
'                  and persistent images.
'-----------------------------------------------------------------
'  Source file:  DETECT.FRM
'  Release version:  1.00                  Programmer:  Lee Adams
'  Type:  Visual Basic global module for Windows applications.
'  Compiler:  Microsoft Visual Basic 1.00
'  Dependencies:  DETECT.BAS global module
'                 DETECT01.BAS module containing Main()
'  Output and features:  Demonstrates a display-independent
'    graphics application that can run in different screen modes.
'    Demonstrates persistent graphics that are automatically
'    refreshed when covered by another window or cropped by the
'    edge of the screen.  Demonstrates how to detect various
'    runtime conditions such as screen resolution, number of
'    available colors, memory mode (real, standard, enhanced).
'  Publication:  Contains material from Windcrest/McGraw-Hill
'    book 4224 published by TAB BOOKS Div. of McGraw-Hill Inc.
'  License:  As purchaser of the book you are granted a
'    royalty-free license to distribute executable files
'    generated using this code provided that you accept the
'    conditions of the License Agreement and Limited Warranty
'    described in the book and on the companion disk.  Government
'    users:  This software and documentation are subject to
'    restrictions set forth in The Rights in Technical Data and
'    Computer Software clause at 252.227-7013 and elsewhere.
'-----------------------------------------------------------------
'      (c) Copyright 1992 Lee Adams.  All rights reserved.
'           Lee Adams(tm) is a trademark of Lee Adams.
'-----------------------------------------------------------------
'  Use the Visual Basic Menu Design Window to set up a menu
'  system with the following attributes:                           (١)
'-----------------------------------------------------------------
'  Menu item              enabled  visible  CtlName     Accelerator
'  &File                     x        x     POPUP_File
'  ----&New                           x     IDM_New
'  ----&Open                          x     IDM_Open
'  ----&Save                          x     IDM_Save
```

```
'  ----Save &As...                           x      IDM_SaveAs
'  ---- -                     x              x      FileSep1
'  ----E&xit                  x              x      IDM_Exit
'  &Edit                      x              x      POPUP_Edit
'  ----&Undo                                 x      IDM_Undo
'  ---- -                     x              x      EditSep1
'  ----Cu&t                                  x      IDM_Cut
'  ----&Copy                                 x      IDM_Copy
'  ----&Paste                                x      IDM_Paste
'  ----&Delete                               x      IDM_Delete
'  &Demo                      x              x      POPUP_Demo
'  ----Create &hidden frame   x              x      IDM_InitFrame
'  ----Create &background     x              x      IDM_DrawBg
'  ----&Clear the display     x              x      IDM_Clear
'  &Help                      x              x      POPUP_Help
'  ----&About                 x              x      IDM_About
'  ----&License               x              x      IDM_License
'  ---- -                     x              x      HelpSep1
'  ----&Screen resolution     x              x      IDM_Display
'  ----Available &colors      x              x      IDM_Colors
'  ----&Memory mode           x              x      IDM_Memory
'  ---- -                     x              x      HelpSep2
'  ----&Help                  x              x      IDM_GeneralHelp   F1
'-----------------------------------------------------------------------
'  Because this Visual Basic application uses a procedure named
'  Main() at startup, you must select Set Startup Form from the
'  Run menu before you run the program or build an .EXE file.

'-----------------------------------------------------------------------
'                     Terminate the application                          (٢)
'-----------------------------------------------------------------------
Sub IDM_Exit_Click ()
If FrameReady = True Then              'if a hidden frame was created
  RetVal = SelectObject(hFrameDC, hPrevFrame)    'deselect bitmap
  RetVal = DeleteObject(hFrame)                  'delete the bitmap
  RetVal = DeleteDC(hFrameDC)          'delete the display-context
End If
End                                              'terminate the program
End Sub

'-----------------------------------------------------------------------
'                   Display the About message box                        (٣)
'-----------------------------------------------------------------------
Sub IDM_About_Click ()
MsgBox "This is a demo from Windcrest McGraw-Hill book 4224.
    ©1992 Lee Adams.  All rights reserved.", MB_OK, "About this
    Visual Basic program"
End Sub

'-----------------------------------------------------------------------
'                 Clear the client area of the window                    (٤)
'-----------------------------------------------------------------------
Sub IDM_Clear_Click ()
zClear                                    'clear the display window
zClearHiddenFrame                          'clear the hidden frame
End Sub

'-----------------------------------------------------------------------
'                  Display the License message box                       (٥)
'-----------------------------------------------------------------------
Sub IDM_License_Click ()
```

```
MsgBox "You can use this code as part of your own software product:
     subject to the License Agreement and Limited Warranty in
     Windcrest McGraw-Hill book 4224 and on its companion disk.",
     MB_OK, "License Agreement"
End Sub

'-------------------------------------------------------------------
'                    Display the Help message box                  (٦)
'-------------------------------------------------------------------
Sub IDM_GeneralHelp_Click ()
MsgBox "Use Demo menu to create a persistent image.  Test by
     moving window off screen or cover with another window.  Use
     Help menu to determine runtime conditions such as resolution,
     available colors, memory mode.", MB_OK, "How to use this demo"'
End Sub

'-------------------------------------------------------------------
'                     Draw a background image                      (٧)
'-------------------------------------------------------------------
Sub IDM_DrawBg_Click ()
'------------------- declare local variables -----------------------
  Dim hPrevBrush As Integer, hSwatchBrush As Integer     'brushes
  Dim hPrevPen As Integer, hBorderPen As Integer            'pens
  Dim X1 As Integer, Y1 As Integer                   'swatch coords
  Dim X2 As Integer, Y2 As Integer                   'swatch coords
  Dim iWidth As Integer, iDepth As Integer       'swatch dimensions
  Dim iSwatchX As Integer, iSwatchY As Integer     'starting coords
  Dim iRed As Integer, iGreen As Integer, iBlue As Integer     'RGB
  Dim iCount As Integer  'loop counter
  Dim iX1 As Integer, iY1 As Integer  'ellipse bounding coords...
  Dim iX2 As Integer, iY2 As Integer
  Dim iGradiant As Integer           'brightness adjustment factor
  Dim iResize As Integer                  'ellipse resizing factor
'------------------ initialize variables ---------------------------
iWidth = zFRAMEWIDE                        'width of a single swatch
iDepth = 4                                 'depth of a single swatch
iSwatchX = 0: iSwatchY = 0                         'swatch location
iX1 = 100: iY1 = 50: iX2 = 300: iY2 = 250           'ellipse coords
iGradiant = 20                             'shading gradiant factor
iResize = 10                                         'sizing factor
'---------------- draw the gradiant background ---------------------
hBorderPen = CreatePen(PS_NULL, 1, RGB(0, 0, 0))        'create pen
hPrevPen = SelectObject(hDC, hBorderPen)  'select transparent pen
X1 = iSwatchX: Y1 = iSwatchY                 'set the swatch coords
X2 = X1 + iWidth: Y2 = Y1 + iDepth    'set swatch width and depth
iRed = 0: iGreen = 0: iBlue = 3                  'set the RGB guns
For iCount = 0 To 63 Step 1                      'for 64 swatches...
  hSwatchBrush = CreateSolidBrush(RGB(iRed, iGreen, iBlue))
  hPrevBrush = SelectObject(hDC, hSwatchBrush)  'select the brush
  RetVal = Rectangle(hDC, X1 - 1, Y1 - 1, X2 + 1, Y2 + 1)   'rect
  RetVal = SelectObject(hDC, hPrevBrush)  'restore default brush
  RetVal = DeleteObject(hSwatchBrush)    'delete the custom brush
  Y1 = Y1 + iDepth                         'reposition top of swatch
  Y2 = Y2 + iDepth                      'reposition bottom of swatch
  iBlue = iBlue + 4                              'increment blue gun
Next iCount
iBlue = 255                                          'reset blue gun
Y2 = 300             'bottom of swatch at bottom of client area
hSwatchBrush = CreateSolidBrush(RGB(iRed, iGreen, iBlue)) 'create
hPrevBrush = SelectObject(hDC, hSwatchBrush)     'select the brush
```

```
RetVal = Rectangle(hDC, X1 - 1, Y1 - 1, X2 + 1, Y2 + 1)      'rect
RetVal = SelectObject(hDC, hPrevBrush)      'restore default brush
RetVal = DeleteObject(hSwatchBrush)      'delete the custom brush
'----------------- draw the gradiant rectangle -----------------
iRed = 0: iGreen = 0: iBlue = 255               'reset the RGB guns
iWidth = 280: iDepth = 4                  'reset dimensions of swatch
iSwatchX = 60: iSwatchY = 24           'reset swatch starting coords
X1 = iSwatchX: Y1 = iSwatchY                  'set the swatch coords
X2 = X1 + iWidth: Y2 = Y1 + iDepth     'set swatch width and depth
For iCount = 0 To 63 Step 1                     'for 64 swatches...
  hSwatchBrush = CreateSolidBrush(RGB(iRed, iGreen, iBlue))
  hPrevBrush = SelectObject(hDC, hSwatchBrush)  'select the brush
  RetVal = Rectangle(hDC, X1 - 1, Y1 - 1, X2 + 1, Y2 + 1)   'rect
  RetVal = SelectObject(hDC, hPrevBrush)   'restore default brush
  RetVal = DeleteObject(hSwatchBrush)    'delete the custom brush
  Y1 = Y1 + iDepth                       'reposition top of swatch
  Y2 = Y2 + iDepth                    'reposition bottom of swatch
  iBlue = iBlue - 4                            'decrement blue gun
Next iCount
'----------------- draw the gradiant circle --------------------
iRed = 0: iGreen = 0: iBlue = 31                'reset the RGB guns
For iCount = 1 To 9 Step 1                      'for 9 ellipses...
  hSwatchBrush = CreateSolidBrush(RGB(iRed, iGreen, iBlue))
  hPrevBrush = SelectObject(hDC, hSwatchBrush)  'select the brush
  RetVal = Ellipse(hDC, iX1, iY1, iX2, iY2)  'draw filled ellipse
  iBlue = iBlue + iGradiant                'increment the blue gun
  iX1 = iX1 + iResize  'adjust the coords for the next ellipse...
  iY1 = iY1 + iResize
  iX2 = iX2 - iResize
  iY2 = iY2 - iResize
  RetVal = SelectObject(hDC, hPrevBrush)   'restore default brush
  RetVal = DeleteObject(hSwatchBrush)         'delete custom brush
Next iCount
'---------------------- tidy up and exit ----------------------
RetVal = SelectObject(hDC, hPrevPen)         'restore default pen
RetVal = DeleteObject(hBorderPen)          'delete the custom pen
zCopyToFrame                             'copy image to hidden frame
End Sub

'--------------------------------------------------------------
'                    Create the hidden frame                     (٨)
'--------------------------------------------------------------
Sub IDM_InitFrame_Click ()
If FrameReady = True Then        'if hidden frame already created
  MsgBox "The hidden frame has already been created.", MB_OK,
    "Graphics system ready"
  Exit Sub
End If
RetLong = GlobalCompact(-1)          'maximize contiguous memory
hFrameDC = CreateCompatibleDC(hDC)          'get a display-context
hFrame = CreateCompatibleBitmap(hDC, zFRAMEWIDE, zFRAMEHIGH)
If hFrame = Null Then                         'if error occurred
  MsgBox "Insufficient memory.  Hidden frame not created.", MB_OK,
    "Graphics system not ready"
  FrameReady = False
  Exit Sub
End If
hPrevFrame = SelectObject(hFrameDC, hFrame)     'select the bitmap
FrameReady = True                              'set a global token
zClearHiddenFrame                          'clear the hidden frame
```

```
MsgBox "Hidden frame successfully created.", MB_OK, "Graphics
    system ready"
End Sub

'--------------------------------------------------------------
'                  Clear the hidden frame                      (٩)
'--------------------------------------------------------------
Sub zClearHiddenFrame ()
If FrameReady = False Then
  Exit Sub
End If
RetVal = PatBlt(hFrameDC, 0, 0, zFRAMEWIDE, zFRAMEHIGH, WHITENESS)
End Sub

'--------------------------------------------------------------
'                 Clear the display window                     (١٠)
'--------------------------------------------------------------
Sub zClear ()
RetVal = PatBlt(hDC, 0, 0, zFRAMEWIDE, zFRAMEHIGH, WHITENESS)
End Sub

'--------------------------------------------------------------
'          Copy the hidden frame to the display window         (١١)
'--------------------------------------------------------------
Sub zCopyToDisplay ()
If FrameReady = False Then
  Exit Sub
End If
RetVal = BitBlt(hDC, 0, 0, zFRAMEWIDE, zFRAMEHIGH, hFrameDC, 0, 0,
    SRCCOPY)
End Sub

'--------------------------------------------------------------
'          Copy the display window to the hidden frame         (١٢)
'--------------------------------------------------------------
Sub zCopyToFrame ()
If FrameReady = False Then
  Exit Sub
End If
RetVal = BitBlt(hFrameDC, 0, 0, zFRAMEWIDE, zFRAMEHIGH, hDC, 0, 0,
    SRCCOPY)
End Sub

'--------------------------------------------------------------
'             Refresh the client area if uncovered             (١٣)
'--------------------------------------------------------------
Sub Form_Paint ()   'is automatically called by Windows as needed
zCopyToDisplay              'copy hidden frame to display window
End Sub

'--------------------------------------------------------------
'            Intercept any attempt to resize the window        (١٤)
'--------------------------------------------------------------
Sub Form_Resize ()        'is called twice when window is resized

If StartUp = True Then  'if window being displayed for first time
  StartUp = False
  IgnoreRefresh = True
  Exit Sub
End If
Form1.WindowState = 0
```

```
                                              'reset normal size
Form1.Width = Window_Width * HorizTwipsPixel          'reset width
Form1.Height = Window_Height * VertTwipsPixel         'reset height
Form1.Left = (Screen.Width - Form1.Width) / 2  'horizontal center
Form1.Top = (Screen.Height - Form1.Height) / 2   'vertical center
If IgnoreRefresh = False Then                      'if second call
  IgnoreRefresh = True          'reset token for next first call
  MsgBox "This demonstration program intercepts any attempt to
    resize the window.  See Windcrest McGraw-Hill book 4224 if you
    want to enable resizing.", MB_OK, "Window size"
  Exit Sub
End If
Form1.Refresh                               'refresh the client area
IgnoreRefresh = False       'if first call, reset for second call
End Sub

'-------------------------------------------------------------
'                Determine runtime memory mode                 (١٥)
'-------------------------------------------------------------
Sub IDM_Mode_Click ()
  Dim TempVariable As Long
TempVariable = MemoryMode And WF_ENHANCED     'perform bitwise AND
If TempVariable = WF_ENHANCED Then     'if result matches constant
  MsgBox "Running in enhanced mode.  Can allocate up to 16 MB
    extended memory (XMS) if available.  Virtual memory up to 4
    times physical memory (maximum 64 MB) is also available via
    automatic disk swapping of 4K pages.", MB_OK, "Memory mode"
  Exit Sub
End If
TempVariable = MemoryMode And WF_STANDARD
If TempVariable = WF_STANDARD Then
  MsgBox "Running in standard mode.  Can allocate up to 16 MB
    extended physical memory (XMS) if available.", MB_OK, "Memory
    mode"
  Exit Sub
End If
MsgBox "Running in real mode.  Can allocate blocks of memory from
    the first 640K of RAM.  Can also allocate blocks from expanded
    memory (EMS) if available.", MB_OK, "Memory mode"
End Sub

'-------------------------------------------------------------
'             Determine number of available colors             (١٦)
'-------------------------------------------------------------
Sub IDM_Colors_Click ()
If DisplayBits = 1 Then                     'if 1 bit-per-pixel...
  If DisplayPlanes = 4 Then                     'if 4 bitplanes...
    MsgBox "Running in 16-color mode.", MB_OK, "Available colors"
    Exit Sub
  End If
  If DisplayPlanes = 1 Then                     'if 1 bitplane...
    MsgBox "Running in 2-color mode.", MB_OK, "Available colors"
    Exit Sub
  End If
End If
If DisplayBits = 8 Then                    'if 8 bits-per-pixel...
  MsgBox "Running in 256-color mode.", MB_OK, "Available colors"
  Exit Sub
End If
MsgBox "Running in a custom color mode.", MB_OK, "Available colors"
End Sub
```

```
'------------------------------------------------------------
'              Determine the screen resolution                     (١٧)
'------------------------------------------------------------
Sub IDM_Display_Click ()
If DisplayWidth = 640 Then
  If DisplayHeight = 480 Then                              'VGA mode
    MsgBox "Running in 640x480 mode.", MB_OK, "Screen resolution"
    Exit Sub
  End If
  If DisplayHeight = 350 Then                              'EGA mode
    MsgBox "Running in 640x350 mode.", MB_OK, "Screen resolution"
    Exit Sub
  End If
  If DisplayHeight = 200 Then                              'CGA mode
    MsgBox "Running in 640x200 mode.", MB_OK, "Screen resolution"
    Exit Sub
  End If
End If
If DisplayWidth = 800 Then             'SuperVGA, 8514/A, XGA mode
  MsgBox "Running in 800x600 mode.", MB_OK, "Screen resolution"
  Exit Sub
End If
If DisplayWidth = 1024 Then                        '8514/A, XGA mode
  MsgBox "Running in 1024x768 mode.", MB_OK, "Screen resolution"
  Exit Sub
End If
If DisplayWidth = 720 Then                           'Hercules mode
  MsgBox "Running in 720x348 mode.", MB_OK, "Screen resolution"
  Exit Sub
End If
MsgBox "Running in custom mode.", MB_OK, "Screen resolution"
End Sub
```

بقية شكل (٣-١٩) قائمة ملف وحدة الشكل لتطبيق العينة

(١) نستعمل نافذة تصميم قائمة بيسيك المرئية لتهيئة نظام بالخواص التالية:

(٢) إنتهاء التطبيق

(٣) عرض صندوق رسالة About (٤) إخلاء منطقة العميل لويندوز

(٥) عرض صندوق رسالة الترخيص (License)

(٦) عرض صندوق رسالة Help (٧) رسم صورة للخلفية

(٨) إنشاء الاطار المختفى

(٩) إخلاء الاطار الحقيقى

(١٠) إخلاء نافذة العرض

(١١) نسخ الاطار المختفى لنافذة العرض

(١٢)نسخ نافذة العرض للاطار المختفى .

(١٣) إنعاش منطقة العميل إذا كانت غير مغطاة .

(١٤) مقاطعة أى محاولة لإعادة تحجيم (to resize) النافذة

(١٥) تحديد طور ذاكرة زمن التشغيل (١٦) تحديد عدد الألوان المتوفرة

(١٧) تحديد دقة الشاشة

تشكيل نظام القائمة :

شفرة التعليق من السطور 0033 إلى 0063 تشرح كيف تشكل نظام القائمة باستعمال نافذة تصميم قائمة البيسيك المرئية (Visual Basic Menu Design Window) ويجب أن نعطى أسماء تحكمCtlname تماماً كما هى مدونة فى القائمة لأن أسماء إجراء الحدث مؤسسة على أسماء التحكم هذه .

وعندما ينتقى المستعملHelp من قائمة Help فى وقت التشغيل ، فمثلاً ، فإن بيسيك المرئية تدعو إجراء حدث يسمى IDM- General Help - CLick () . إجراء الحدث هذا موجود عند السطر0103 فى القائمة . وإذا أعطينا إسم تحكم مختلف ، لبند قائمة Help، فإن بيسيك المرئية لن تدعو:- IDM - General Help Click ( )

الحروف الأولى (Prefixes) - POPUP , IDM فى قائمة CtlName تستعمل هنا حتى تعطى إتساق بالتقليديات المستعملة بواسطة طاقم تطوير برنامج ميكروسوفت ويندوز . ويمكننا تقريباً أن نستعمل أى إسم تحكم نريده ، بشرط أن نسمى بالمثل إجراء الحدث المقابل . فمثلاً :

يلاحظ أن بعض بنود القائمة لم تدعم . بند قائمة The & Open فى قائمةThe & File ، لم يتم تدعيمه إذا كان المستخدم ينتقى بند قائمة Open عند زمن التشغيل ، فإن بيسيك المرئية لن تعمل شيئاً . إذا كان بند قائمة Open & مدعماً ، ببساطة فإن بيسيك المرئية ستدعو روتين يعيد التحكم للداعى (donothing Stub) يسمى( ) IDM- Open - Click والذى يحتوى على عبارات Sub، End فقط . وبالعكس ، فإن بند قائمة Create، background فى قائمة The & Demo هى مثال جيد لبند قائمة مدعم .

وعندما ينتقى المستخدم بند القائمة هذا فى زمن التشغيل ، فإن بيسيك المرئية ستدعوإجراء حدث (event procedure) يسمى IDM - Draw Bg - Click()

والذى وجد فى السطر0110 فى قائمة البرنامج .

ويلاحظ السطر0063 بالأخص حيث تدون F1 كمعجل . وأثناء ما نختبر تطبيق العينة ، يمكننا ضغط F1 ، وسيظهر صندوق رسالة Help .

إنشاء الإطار المختفى :

إجراءات الأحداث فى السطور من 0193 إلى 0210 تنشىء bitmap المختفية . عبارة 1F عند السطور 0194 إلى 0197 تفحص متغير يسمى Frame Ready لتأكيد أن bitmap المختفية لم تنشأ بعد . الدعوة لوظيفة-GDI's Global Com pact () عند السطر 0198 تجبرويندوزعلى أداء نوع ذاكرة ، لتنشىء كثيراً من ذاكرة قريبة بقدر الامكان . الشفرة عند السطور 0199 إلى 0200 حينئذ تهيىء محيط/ عرض وتنشىء محيط/ عرض ذاكرة bitmap ، وتنشىء متسقة . قالب1F عند السطور 0201 إلى 0205 توزع تحذير وتلغى تشغيل إذا كانت الدعوة لـCreate Compatible Bitmap ( ) أرجعت مقبض غير صحيح .

الشفرة عند 0206 تنتقىbitmap الذى أنشىء حديثاً داخل محيط/ عرض الذاكرة عند زمن التشغيل توكين يسمى Frame Ready يهيأ إلى True عند السطر0207 . السطر0208 يدعو إجراء (عند السطر0215) لإخلاء الاطار المختفى. أخيراً ، صندوق رسالة يبلغ الإنشاء الناجح لـ bitmap المختفى.

إخلاء نافذة العرض :

الإجراء الوظيفى عند السطور 0225 إلى0227 يخلى نافذة العرض بدعوة وظيفة GDI,s Bat Blt ( ) .الثوابتzFRAMEWINDOW ، zFRAMEHIGH كانت قد أعلنت عند السطور 0049 إلى 0050

فى وحدة detect. bas الشاملة WHITENESS . الثابت هو ثابت ظهور ويندوز معرف عند السطر0084 فى detect. bass الوحدة الشاملة . إجراء وظيفة

the zClear () عند السطر0225 يدعى بواسطة إجراء حدث ()- IDM - Clear Click عند السطور 0088 إلى 0091.

إخلاء الإطار الخالى :

الإجراء الوظيفى عند السطور 0215 إلى 0220 يخلى الاطار المختفى بدعوة Pat .Bit()

نسخ الإطار المختفى :

الإجراء الوظيفى عند السطور 0232 إلى 0237 تنسخ bitmap المختفية لنافذة العرض . يلاحظ كيف يفحص السطر0233 لتأكيد أن bitmap يتواجد قبل محاولة نسخه . دعوة () The BitBlt عند السطر0236 تنسخ الـ bitmap خاصيةhDC المزودة بواسطة بيسيك المرئية هى جهة الوصول :hFrame DC هو المصدر .

نسخ نافذة العرض :

الإجراء الوظيفى عند السطور 0242 إلى 0247 ينسخ منطقة العميل لنافذة العرض لـ bitmap المختفية. ويعمل بنفس الطريقة مثل الشفرة عند السطور0232 إلى0237 .

رسم الصورة :

إجراءات الحدث عند السطور0110 إلى 0188 ترسم الرسومات منحدرة . (gradient) . الشفرة عند السطور 0112 إلى 0123 تعلن وتبدأ متغيرات محلية .

عبور رقيق بين درجتي لون ، تسمى	gradient
تسلق أيضاً : منحدرة :	

حلقة For Next عند السطور 0137 إلى 0146 ترسم منحدرة اللون الحلقية . الشفرة عند السطر0145 تضبط قيمة القاذف الأزرق على كل مسار خلال الحلقة .

الحلقة عند السطور 0160 إلى 0196 ترسم المستطيل منحدر (gradient)، والحلقة عند السطور 0172 إلى 0183 .

ترسم الدائرة منحدرة (gradient). ويلاحظ كيف أن كل حلقة تشطب أدوات رسمGDI التى تستعملها .

عرض صناديق الرسائل :

الاجراء الحدث () IDM- About - Click عند السطور 0081 إلى 0083 تعرض صندوق رسالة About. عبارة MsgBox لبيسيك المرئية تستدعى عند السطر0082 لادارة الـ Caption ، لـ text (نص)، وزر Ok . لصندوق الرسالة .

( ) The IDM - License - Clickعند السطور 0103 إلى 0105 تعرض صندوق رسالة Help.

الرسومات المتوالية (Persistent) :

إجراء ( ) Form - Paint عند السطور 0252 إلى 0254 تدعى أوتوماتيكياً بواسطة بيسيك المرئية عندما تحتاج منطقة عميل ويندو التطبيق أن تنتعش . وهذا يمكن أن يحدث نتيجة لتحريك النافذة أو إزالة غطائها أو تحريكها خلفياً بعد أن تم قص جزء منه بطرف الشاشة . ويندوز تولد رسالة WM - PAINT والتى تعيد بيسيك المرئية مسارها لاجراء () Form - Paint . ويلاحظ كيف يدعو السطر0523 الاجراء الوظيفى () zCopy To Display الموجود عند سطر0232 لنسخ الصورة من الإطار المختفى لنافذة العرض ، وبذلك تنعش المنطقة الحالية .

عادة، فإن بيسيك المرئية تعتنى كلية بانعاش منطقة العميل للشكل الأساسى ، ولكن هذه الخاصية يتم إعاقتها بعبارة Form 1. Auto Redraw = False عند سطر0095 فى وحدة بدء detect 01. bas شكل (٣ ـ ٢٠) . وللقدرة على إعطاء

عمليات إنعاش يعطى تحكم أكثر على تطبيقنا ، كما هو موضح ببرامج الرسومات المتحركة . شكل (٣ ـ ٢٠) :

```
  VertTwipsPixel = 12!
  Window_Width = zWINDOW_WIDTH
  Window_Height = 362!                       'compensate for font size
End If
If DisplayWidth = 1024 Then                  '8514/A, XGA 1024x768 mode
  HorizTwipsPixel = 12!                     '12200x9216 twips full screen
  VertTwipsPixel = 12!
  Window_Width = zWINDOW_WIDTH
  Window_Height = 362!                       'compensate for font size
End If
If DisplayWidth = 720 Then                        'Hercules 720x348 mode
  HorizTwipsPixel = 12!
  VertTwipsPixel = 20!
  Window_Width = zWINDOW_WIDTH
  Window_Height = 342!      'adjust for aspect ratio and font size
End If
'------------- initialize and display the window ---------------
Form1.Width = Window_Width * HorizTwipsPixel            'set width
Form1.Height = Window_Height * VertTwipsPixel          'set height
Form1.Left = (Screen.Width - Form1.Width) / 2  'horizontal center
Form1.Top = (Screen.Height - Form1.Height) / 2   'vertical center
Form1.Caption = "Display-independent graphics"   'set the caption
Form1.AutoRedraw = False                 'disable automatic refresh
Form1.BackColor = RGB(255, 255, 255)   'set the client area color
Form1.ForeColor = RGB(0, 0, 0)                      'active color
Form1.ScaleMode = PIXELS                    'will use pixel coords
Form1.Show                                    'display the window
'-------------------- display startup text ---------------------
Form1.CurrentX = 10: Form1.CurrentY = 282        'set text coords
PreviousColor = Form1.ForeColor            'remember current clr
Form1.ForeColor = RGB(191, 191, 191)                     'set clr
Form1.Print "- 1992 Lee Adams.  All rights reserved." '- Alt+0169
Form1.ForeColor = PreviousColor           'restore default clr
'------------------- check if mouse present --------------------
MousePresent = GetSystemMetrics(SM_MOUSEPRESENT)
If MousePresent = 0 Then                              'if no mouse
  Beep
  MsgBox "No mouse found.  Some features of this demo program may
    require a mouse.  The demo's menu system also responds to the
    keyboard.  Press <Enter> to continue.", MB_OK, "Graphics
    system warning"
End If
End Sub

'-------------------------------------------------------------
'              Reusable template for startup code for
'            Visual Basic Windows graphics applications.
'-------------------------------------------------------------
'  Source file:  DETECT01.BAS
'  Release version:  1.00                 Programmer:  Lee Adams
'  Type:  Visual Basic startup module for Windows applications.
'  Compiler:  Microsoft Visual Basic 1.00
'  Dependencies:  DETECT.BAS global module
'                 DETECT.FRM primary module
'  Output and features:  Initializes the runtime environment
```

```
'    for a Windows graphics application created with.
'    Visual Basic.  Ensures runtime image size and compatibility
'    no matter which graphics mode is being used by Windows.
' Publication:  Contains material from Windcrest/McGraw-Hill
'    book 4224 published by TAB BOOKS Div. of McGraw-Hill Inc.
' License:  As purchaser of the book you are granted a
'    royalty-free license to distribute executable files
'    generated using this code provided that you accept the
'    conditions of the License Agreement and Limited Warranty
'    described in the book and on the companion disk.  Governmen
'    users:  This software and documentation are subject to
'    restrictions set forth in The Rights in Technical Data and
'    Computer Software clause at 252.227-7013 and elsewhere.
'------------------------------------------------------------
'      (c) Copyright 1992 Lee Adams.  All rights reserved.
'          Lee Adams(tm) is a trademark of Lee Adams.
'------------------------------------------------------------

'------------------------------------------------------------
'       Initialization code for startup of application
'------------------------------------------------------------
Sub Main ()                           'is called by Windows at startup
  Dim PreviousColor As Long          'will remember default color
StartUp = True    'set runtime tokens used by other procedures...
IgnoreRefresh = True
FrameReady = False
'---------------- examine the graphics adapter -----------------
hDesktopWnd = GetDesktopWindow()              'grab handle to desktop
hDCcaps = GetDC(hDesktopWnd)       'get display-context for desktop
DisplayWidth = GetDeviceCaps(hDCcaps, HORZRES)  'horiz resolution
DisplayHeight = GetDeviceCaps(hDCcaps, VERTRES)  'vert resolution
DisplayBits = GetDeviceCaps(hDCcaps, BITSPIXEL)   'bits-per-pixel
DisplayPlanes = GetDeviceCaps(hDCcaps, PLANES)  'num of bitplanes
RetVal = ReleaseDC(hDesktopWnd, hDCcaps) 'release display-context
'--------------- determine the runtime memory mode -------------
MemoryMode = GetWinFlags()     'will decipher this variable later
'--------------- set mode-dependent twips factors --------------
HorizTwipsPixel = 15!: VertTwipsPixel = 15!      'set defaults...
Window_Width = zWINDOW_WIDTH: Window_Height = zWINDOW_HEIGHT
If DisplayWidth = 640 Then
  If DisplayHeight = 480 Then                      'VGA 640x480 mode
    HorizTwipsPixel = 15!          '9600x7200 twips full screen
    VertTwipsPixel = 15!
    Window_Width = zWINDOW_WIDTH
    Window_Height = zWINDOW_HEIGHT
  End If
  If DisplayHeight = 350 Then                      'EGA 640x350 mode
    HorizTwipsPixel = 15!          '9600x7000 twips full screen
    VertTwipsPixel = 20!
    Window_Width = zWINDOW_WIDTH
    Window_Height = 342!   'adjust for aspect ratio and font size
  End If
  If DisplayHeight = 200 Then   'nominal support CGA 640x200 mode
    HorizTwipsPixel = 15!
    VertTwipsPixel = 36!
    Window_Width = zWINDOW_WIDTH
    Window_Height = zWINDOW_HEIGHT
```

```
0069      End If
0070    End If
0071    If DisplayWidth = 800 Then      'SuperVGA, 8514/A, XGA 800x600 mode
0072      HorizTwipsPixel = 12!               '9600x7200 twips full screen
```

شكل (٣-٢٠) : قائمة ملف وحدة البدء لتطبيق العينة

إعادة تحجيم (Resizing) النافذة :

إجراء () Form - Reseize عند السطر 0259 يدعى أوتوماتيكياً بواسطة بيسيك المرئية عندما تكتشف محاولة من المستخدم لاعادة الاستيلاء على ويندو فى وقت التنفيذ. ويلاحظ كيف أن الشفرة عند السطور 0265 إلى 0269 تعيد تهيئة خواص متعددة لـ Forml الشكل الأساسى لاستعادة الاستيلاء الأصلى. حينئذ ، فإن العبارة 1F عند سطر 0270 تفحص علم زمن تشغيل المسمى Ignore Refresh لتقرر إذا كان صندوق رسالة تم عرضه . هذا العلم يستفيد من ميزة خاصية (idiosyncrasy) فى المشترك بين بيسيك المرئية ووبندوز والذى يجعل ()- Form Resize أن يدعى مرتين عند عمل محاولة لاعـادة الاستيلاء على النافـذة. أخيراً ، يلاحـظ كيـف أن سطر 0225 يستعمل خاصيـة Refresh (الانعاش) للشكل لاجبار دعوة إلى إجراء () Form - Paint .

ننظر للجزئين السابقين للكلام عن الاجراء The Form - Paint .

طور ذاكرة زمن التشغيل :

إجراء حدث () IDM - Mode - Click عند السطور 0282 إلى 0295 تبلغ إذا كانت ويندوز تنفذ فى طور عيارى أو طور مدعم . ويعرض صندوق الرسالة لكل طور يصف القدرات لطور الذاكرة المستعمل بواسطة ويندوز . ويلاحظ أن الشفرة عند السطر 0294 هى النص الذى يستعمل كثيراً بدون تغيير (boilerplate) والذى لن ينفذ بتاتاً لأن إصدار التحرير لبيسيك المرئية ينشىء تطبيقات والتى تحتاج طور إما عيارى أو مدعم .

الألوان المتوفرة :

إجراء حدث () IDM - Mode - Click عند السطور 0300 إلى 0316 تبلغ أقصى عدد من الألوان يمكن عرضه فى نفس الوقت . المتغيرات المسماه Display Bit ،Display Planes كانت بادئة بواسطة وحدة بدء detect 01 - bas ونتكلم عنها فى هذا الجزء .

خواص الأجزاء الصلبة (موائم رسومات الخواص) تستعمل لتظهر قدرات اللون القصوى لطور الرسومات الحالى .

دقة الشاشة :

إجراء حدث() IDM - Display - Click عند السطور 0321 إلى 0340 يبلغ دقة الشاشة . المتغيرات المسماة Display Widh ، Display Height تم بدءها بواسطة وحدة بدء detect 01. bas

إنهاء التقرير :

إجراء الحدث عند السطور 0069 إلى 0076 ترتب بيئة GDI قبل توزيع عبارة End. ويلاحظ كيف أن عبارة IF عند السطر 0070 تفحص علم حالة (status) لتحديد إذا كان إطار مختفى كان قد أنشأ . إذا كان كذلك، يجب أن يكون bitmap غير منتقى (السطر 0071) قبل أن يشطب (السطر 0072) . حينئذ ، السطر 0073 يشطب محيط/ عرض (display - Context) .

كيف تعمل وحدة البدء (Startup) :

القائمة detect 01. bas لها وحدة بدء موجودة فى شكل (٣ ـ ٢٠) الاجراء المسمى ()Main فى هذه الوحدة، يتم دعوته عندما تطلق ويندوز التطبيق ـ نرجع لشكل (٢ ـ ٦) ، شكل (٢ ـ ٧) .

قدرات العرض :

الدعوة لوظيفة ويندوز API ( ) Get Device Caps عند السطور 0041 إلى 0043 تستعمل لاسترجاع القيم التى ستستعمل عندما يريد المستخدم فحص دقة الشاشة والألوان المتوفرة . ويلاحظ الثوابت :,HORZRES, VERTRES BITSPIX & PLANES ، والتى كانت أعلنت فى وحدةdetect.bas الشاملة فى توقع هذه الدعوات لـ () Get Device Caps. ويلاحظ أيضاً كيف أن دعوة إلى( ) API,s Get Desktop Window عند السطر0039 تستعمل لاستجلاب المقبض للشاشة كلها . حينئذ، فإن الشفرة عند السطر0040 تدعو وظيفة ()GDI,s Get DC لتهيئة محيط/ عرض للشاشة .

طور الذاكرة : (Memory Mode)

الشفرة عند السطر0047 تدعو وظيفة () API,s GetWin Flags والتى ترجع مجموعة من الأرقام الثنائية والتى تصف الخواص لنظام زمن التشغيل (runtime) . المتغير المسمى (Memory Mode) يستعمل بواسطة إجراء الحدث (event pro- cedure):

IDM- Mode - Click ( ) فى وحدة detect. Frm حتى تبلغ طور ذاكرة زمن التشغيل (عيارى أو مدعم) .

تبعية العرض : (Display - Dependency)

الشفرة عند السطور 0049 إلى 0088 تعمل ضبط للتعويض عن الطريقة التى تتناول فيها البيسيك المرئية دقات مختلفة للشاشة .

البيسيك المرئية تستعمل(twips) لتعريف قياسات على الشاشة .twip عبارة عن $\frac{1}{1440}$ من البوصة . وهذا يعنى أن طور 640 x 480 VGA عبارة عن 9600 x 7000 twips. ويلاحظ كيف تهيىء الشفرة قيم متغيرين يسميانHoriz

Twips Pixel & VertTwipsPixel . هذه القيم تستعمل بعد ذلك فى السطور 0091،0090 لتهيئة أبعاد الشكل الأساسى (نافذة عرض التطبيق: -The ap) plication's display Window

وحدة قياس : twip ، وتساوى $\frac{1}{1440}$ من البوصة، أو $\frac{1}{20}$ من النقطة وتستعمل فى التطبيقات .

بدء الويندو :

الشفرة عند السطور 0090 إلى 0091 تبدأ وتعرض ويندو التطبيق . السطور 0090 إلى 0091 تعرف عرض وارتفاع الويندو . السطور 0092 إلى 0093 تحسب موضع التمركز الأفقى والرأسى ، لتأكيد أن الويندو تظهر فى منتصف الشاشة ، ولا يهم أى طور رسومات يتم استعماله .

خاصية التعليق (Caption) على الشكل تعرف فى السطر 0094 . لون الخلفية ولون الأمامى وخواص مقياس رسم القياس تهيأ بواسطة السطور0096 ،0097 ،0098 . وتوظف طريقة Show عند السطر0099 لعرض الويندوعلى الشاشة .

ويلاحظ بالأخص ، سطر0095 والذى يهيىء خاصية Auto Redraw للشكل الأساسى إلى False ، مؤكداً أن أجزاء () Form - Paint فى وحدةFrm . detect تدعى عندما تحتاج ويندو أن تنتعش .

اكتشاف الفأرة :

الشفرة عند السطور0107 إلى0111 تكشف إذا كانت فأرة مؤثرة موجودة . السطر0107 يدعو وظيفة () GDI's Get System Metrics مع ثابت- SM MOUSEPRESENT (والذى كان أعلن فى وحدة detect. bas الشاملة) . عبارة بيسيك المرئية MsgBox فى السطر0110 توزع تحذير استشارى (advisory) إذا لم توجد أى فأرة. تطبيق العينة سيستمر فى التنفيذ بطريقة صحيحة بدون فأرة . بعض

تطبيقات الرسومات ، تعتمد على دخل الفأرة ، والشفرة عند السطور 0107 إلى 0111 هي طريقة جيدة لنا لاختبار وجود فأرة عند بدء البرنامج .

تدوين البرنامج لتطبيق العينة :

تدوينات البرنامج المقدمة هنا تحتوى على كل شفرة المصدر التى تحتاج لبنائها وتنفيذ تطبيق العينة . التدوين للوحدة الشاملة موجود فى شكل (٣ ـ ١٨) . والتدوين لوحدة الشكل (Form module) يوجد فى شكل (٣ ـ ١٩) ، بينما التدوين لوحدة البدء ففى شكل (٣ ـ ٢٠).

القرص المرافق :

إذا كان لدينا القرص رفيق (Companion) ، فإن ملفات المصدر تقدم : مثل : detect. mak. detect. bas. detect. frm & detect O. bas.

الترخيص(License) :

يمكننا استرداد مقبض من قلب ذاكرة ويندوز (paste) :شفرة المصدر المعفاة من الضرائب (royalty - Free) فى تطبيقنا وتوزيع الملفات المنفذة الناتجة تحت ظروف License Agreement & Limited Warranty فى مقدمة هذا الكتاب .

الباب الرابع

الشـــخصيات

(Characters)

الباب الرابع
الشخصيـــــات
(Characters)

رسوم الرموز المتحركة هى واحدة من أكثر مظاهر التحدى لرسومات الحاسب الآلى المتحركة. كذلك ، فهى واحد من الأكثر قبولاً والمجزية . سواء كانت الشخصية (realistic) ، أو كاريكاتيرية (Caricature) أو (Cartoon) فلا تعمل فرق للتأثير (impact) الكبير والذى يمكن لرسوم لرموز المتحركة أن تجعله على عرض حاسب آلى ..عملاؤنا، وزبائننا ومستخدمى الطرفيات سيكونون مسحورين ومأخوذين، فهى ستعطى برنامجنا الاهتمام الغير منحرف أثناء تتابعات رسوم الرموز المتحركة . يوجد شىء لا يقاوم عن الوجه الأدمى والذى يفتن متفرج. حتى متفرج واحد .

رسوم الرموز المتحركة تحتاج أن تتحرك الرموز. تصميم ورسم هذه الشخصات هو عمل عظيم والذى يحتاج لتخطيط حريص وعمل شاق ، حتى إذا كنا فنانين طبيعين أو حتى أكثر إذا لم نكن كذلك. ولا يهم كيف تكون آلة رسومنا وجيهه، ولا يهم كيف تكون شفرة مصدرنا وجيهه، ولا يهم كيف يكون سجل خواصنا ، فإن نجاح رسومنا المتحركة مربوط بقوة الرمز الذى نرسمه . معرفة كيف نعمل الرسومات ليس كافيا ، فيجب أن نعرف أيضاً ماذا نرسم . الحاسب لآلى وبرنامجنا هما المرحلة التى عليها سيؤدى الرمز. ببساطة ، يجب أن يكون لدينا رسم جودة حرفية للرمز الذى نخطط لرسمه .

تصميم وإنشاء ورسم رموز لرسومات الحاسب الآلى المتحركة هو جزء قابل

للأدارة لعملية الانتاج إذا تبنينا إتجاه منهجى . حتى إذا كان لدينا قليلا أو حتى عدم وجود خبرة فنية ، فان هذا الباب سيعلمنا :

- كيف نرسم الرؤوس والوجوه
- كيف نرسم الأيدى
- كيف نرسم الأذرع والأرجل والأجسام
- كيف نرسم (animate) الأعين
- كيف نرسم الحديث والحوار
- كيف نرسم تعبيرات وجهيه
- كيف نرسم رأس تدور

فى هذا الباب أيضاً تطبيق عينة والذى يوضح رسم متحرك وجهى (facial) فهو يمثل رمز والذى يعرض مدى لتعبيرات تتراوح بين ابتسامة بسيطة إلى عبوس شيطانى (diabolical scowl)، وتتضمن إندهاش واسع العيون باستعمال أخذ مزدوج عيارى / الصناعة .

كثير من الفوائد والدروس والأسس والتى سنكتشفها فى هذا الباب تبيناها من وسائد الرسومات ، وجداول خفيفة وغرف سائرة لرسومات متحركة لأفلام تقليدية. كثير من المادة مقطرة من وسائل رسم والتى كانت قد قدمت ودققت فى الثلاثينيات والأربعينيات بواسطة موضحين وفنانين عاملين فى صناعة الفن التجارى. جودة لا نهائية (Timelesss) لهذه الوسائل للرسم تم تضيحها بواسطة عدد من الموضحين والفنانين والذين يستمرون فى استعمالها هذه الأيام .

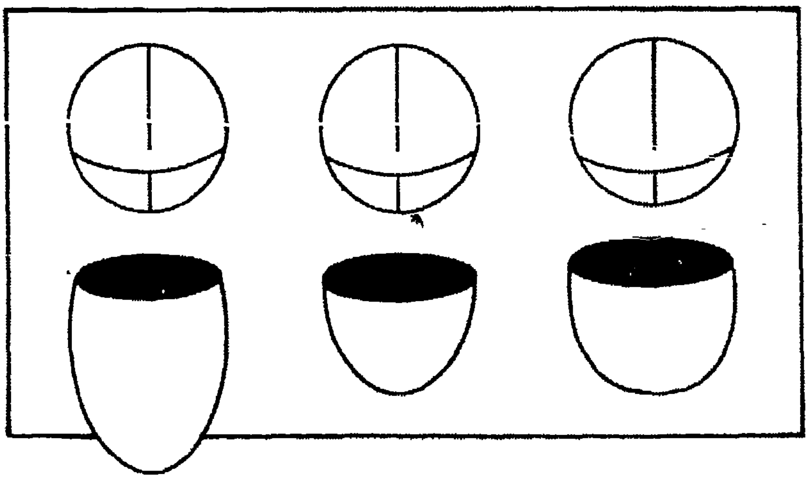
كيفية رسم الرؤوس والوجوه :

القدرة على رسم رؤوس ووجوه بنظرة حرفية هى مهارة صالحة للعرض ذات

طلب كبير ، خاصة رسوم الحاسب الآلى المتحركة . ويمكننا إحكام هذه المهارة باتباع إتجاه يستعمل عناصر لبناء رؤوس ووجوه رموز.

عناصر البناء :

الأمثلة فى شكل (٤ ـ ١) تبين بعض عناصر البنـاء التى نحتاجها لبنـاء رؤوس رموز. فكرة الحاق فـك (a jaw) لجمجمـة كرويـة لبنـاء رأس مثـل الحقيقية تـتـم بواسطة كثير من مدرسى الفن ،والفنانين التجاريين ، ورسامى الرسوم المتحرك . ولكن المفهوم تسلم أولاً تحذير منتشر فى كتاب تعليمات الفن الأسطورى (legendary)/ now) ،(Figure Drawing For All It's Worh باعلان الموضح Andrew Loomis (نشر بمطبعة Viking Press فى عام ١٩٤٣ ، والآن غير مطبوع) . وغيره .

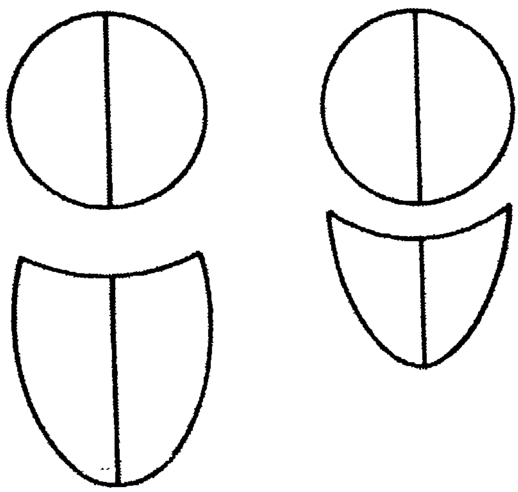


شكل (٤ـ١) : عناصر البناء لرؤوس رموز كارتون ، كما بحثت بواسطة كثير من مدرسى الفن ، والفنانين التجاريين ، ورسامى الكارتون والرسوم المتحركة .

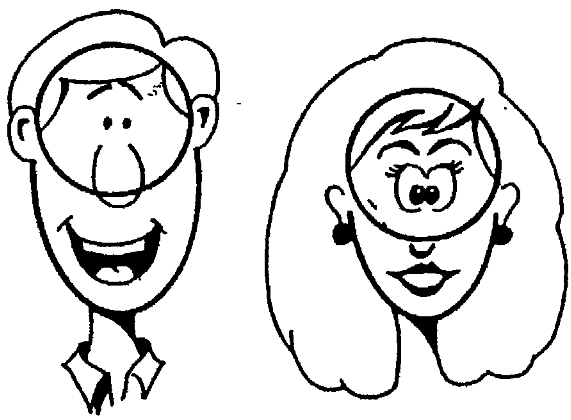
بناء الذكر والأنثى :

يمكن تبنى طريقة عنصر البناء لرموز الذكر والأنثى . الأمثلة فى شكل (٤ ـ ٢)

تبين الفرق بين عناصر بناء الذكر والأنثى . باستعمال فك (jaw) أصغر بنقطة حادة أكثر ، يمكن تطوير رمز أنثى . رموز الذكر والأنثى فى شكل (٤ ـ ٣) تبين كيف تستعمل عناصر البناء فى شكل (٤ ـ ٢) لبناء رموز منتهية . عناصر البناء تؤكد أن لدينا الشكل الأساسى صحيح ، بينما تفاصيل الانهاء مثل العيون والفم والأنف والشعر تحضر الرمز للتكملة .



شكل (٤-٢) : جهة اليسار عناصر بناء لرموز كارتون الذكر . وفى اليمين عناصر بناء لرموز كارتون الأنثى .



شكل (٤-٣) : تفاصيل رموز كارتون الذكر والأنثى، مبينة كيف أن عناصر البناء فى شكل (٤-٢) تستعمل لبناء الرسم المنتهى.

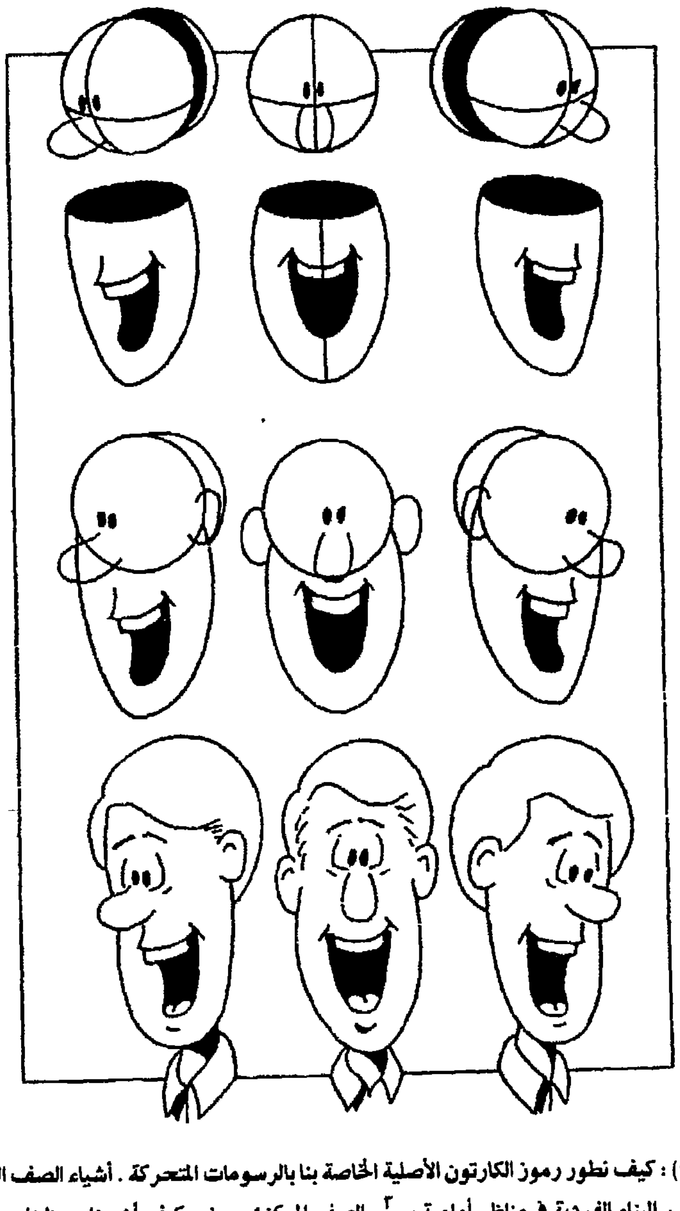
المناظر المختلفة :

قليل من تتابعات الرسوم المتحركة تنتج باستعمال مناظر أمامية من الرموز فقط. لحسن الحظ ، طريقة عنصر البناء تجعل من السهل تصميم وبناء مناظر رمز واحد فى الأمام، وثلاثة أربع ولمحة جانبية صفحة الموديل فى شكل (٤ ـ ٤) يبين كيف تستعمل عناصر البناء لبناء الشخصيات الأصلية الخاصة بنا فى مناظر وأمامية . ويلاحظ مناظر على وجه الخصوص ، حيث تستعمل نصف كرة لتمثيل خلفية الجمجمة هذه المجموعة من عناصر البناء الثلاثة تعطينا الشكل الأساسى (un-derlying) الذى نحتاجه لعمل موديل حد الشعر لشخصياتنا. كذلك يلاحظ كيف أن خطوط الطول والمدى على الكرة فى شكل (٤ ـ ٤) تساعد فى الوضع الصحيح للعيون والأنف . هذا الشكل للفم هو نوع شائع يستعمل كثيراً بواسطة فنانين محترفين. وأخيراً ، فى هذا الباب سنرى كيف نعدل ونتبنى الفم للتحرك للحوار والكلام .

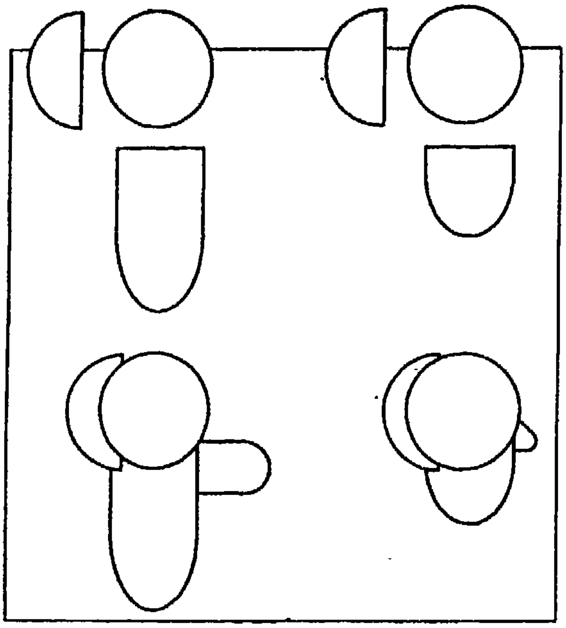
إنشاء منظر جانبى لشخصيتنا سهل . الأمثلة فى شكل (٤ ـ ٥) تبين عناصر بناء لمناظر جانبيه . مرة أخرى ، نلاحظ كيف أن ثلاثة عناصر بناء تستعمل لتمثيل الجمجمة ، وخلفية الجمجمة ، والفك . ويلاحظ كيف أنه أضيفت أنف عند تجميع عناصر البناء الثلاثة . كذلك ، يلاحظ الفك الأصغر والمستعمل فى إصدار الأنثى للمنظر الجانبى فى شكل (٤ ـ ٥) .

أسلوب الانهاء :

يمكن استعمال كثير من الاساليب (Syles) المختلفة لانها شخصياتنا ، يتراوح بين بسيط، وصارم ، Spartan ، والمركب والتفصيلى. الأمثلة فى شكل (٤ ـ ٦) توضح الفرق بين تصميم بسيط ورؤوس شخصيات ذات تفاصيل عالية . التصميمات البسيطة أسهل أحيانًا فى التقوية أو التشجيع ، ولكن التصميمات التفصيلية تعطى غالبا ، أصيلة أكثر وتتابعات رسوم متحركة نابضة بالحياة (life like) .

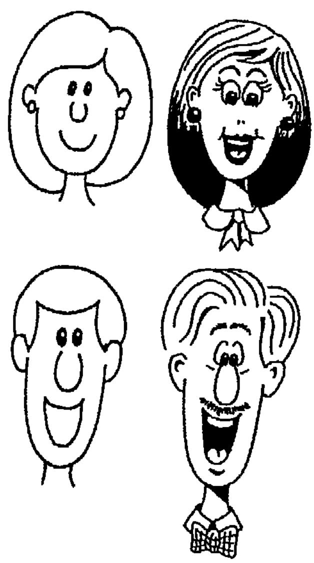


شكل (٤-٤) : كيف نطور رموز الكارتون الأصلية الخاصة بنا بالرسومات المتحركة . أشياء الصف العلوى ترسم عناصر البناء الفردية فى مناظر أمامية ، $\frac{٣}{٤}$ الصف المركزى يوضح كيف أن عناصر البناء ملحقة ببعضها لعمل الشكل الأساسى لرأس شخصية الكارتون الصف الأسفل يبين تفاصيل البناء لمناظر $\frac{٣}{٤}$ والأمامى .



شكل (٤-٥) : عناصر بناء لمناظر جانبية لشخصيات كارتون . نوع ذكر مبين جهة اليسار . إصدار الأنثى مبين فى اليمين .

ولا يهم أى اتجاه نتخذه لانهاء تصميمات الشخصيات ، ونتذكر فقط أن نبدأ دائماً بعناصر البناء . فهى ضماننا لنتائج حرفية .

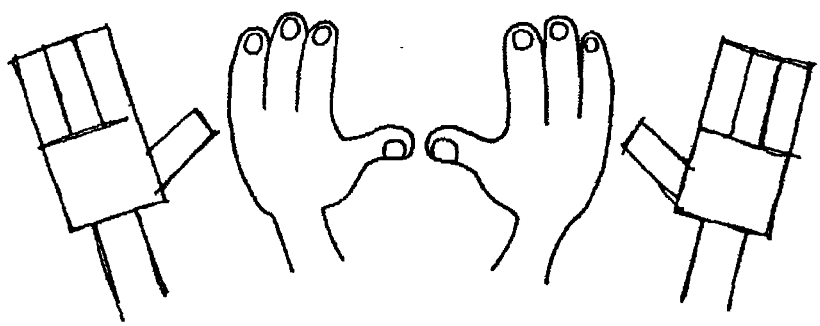


شكل (٤-٦) : كمية التفصيلات فى رسم تؤثر على الادراك الحسى للمشاهد وكمية وقت الانتاج المطلوبة .

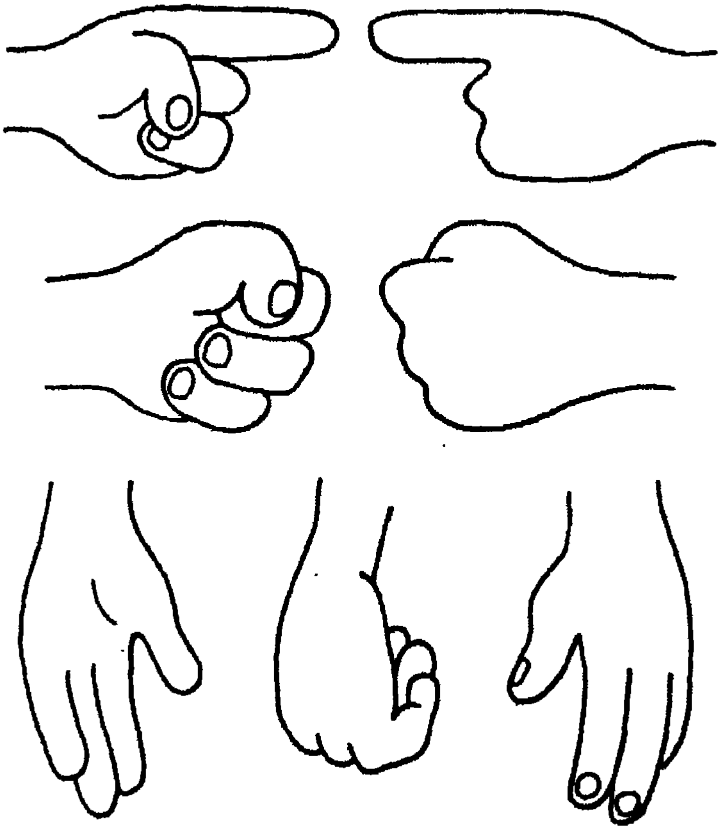
كيفية رسم الأيدى :

تعلم رسم الأيدى أكثر صعوبة من رسم الرؤوس والأوجه إذا استعملنا اتجاه البناء. الأمثلة فى شكل (٤ ـ ٧) تبين كيف نستعمل عناصر البناء لعمل الأيدى . أولاً ، نجمع يد بخطوط مستقيمة (box - like) ، ثم نضيف المنحنيات التى تعطيها الحياة . الأمثلة فى شكل (٤ ـ ٨) توضح أنواع من مواضع اليد المستعملة ، وكل منها كانت منشأة من نموذج أصلى (prototype) مكون من عناصر بناء .

إذا كان تتابع الرسوم المتحركة يستعمل شخصيات كاريكاتيرية أو كارتون ، حينئذ يجب أن تستعمل عيارى / الصناعة ، يد ذات ثلاثة أصابع والمبينة فى شكل (٤ ـ ٧)، وشكل (٤ ـ ٨) إذا كانت شخصيتنا واقعية (realistic) أو نصف واقعية نرغب فى الأخذ فى الاعتبار يد ذات الأربعة أصابع ، والتى تحاكى الحياة ، ولكن أصعب أكثر فى الرسم وعمل الرسوم المتحركة .



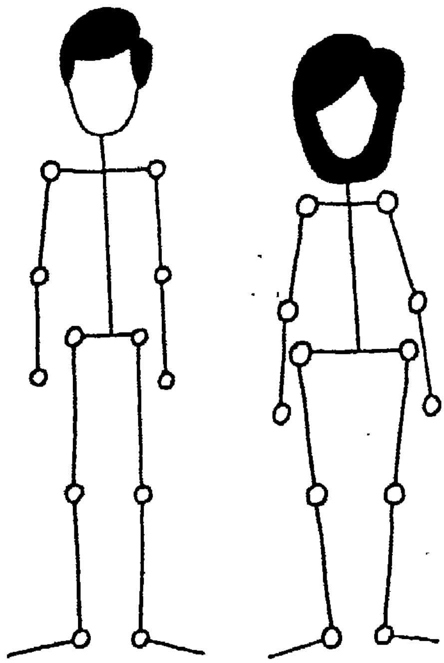
شكل (٤ ـ ٧) : عناصر بناء للايدى أو لشخصيات كارتون لرسوم متحركة .



شكل (٤-٨) : عينات أوضاع اليد ، وتفيد فى رسومات الحاسب الآلى المتحركة لشخصيات الكارتون .

كيفية رسم الأذرع والسيقان والأجسام :

الأمثلة فى شكل (٤ ـ ٩) يبين كيف نستعمل عناصر بناء Stick لبناء الأذرع ، والسيقان و الجذوع (torsos) . ويلاحظ كيف أن شكل الأنثى ينشأ بأرداف أعرض وأكتاف أضيق عن شكل الذكر . شكل الأنثى أقصر ككل ، مع ظهرها أقصر كثيرًا عن الذكر ، ولكن سيقانها تكون بنفس الطول تقريبا مثل شكل الذكر .



شكل (٤-٩) : عناصر بناء لشخصيات رسوم متحركة .

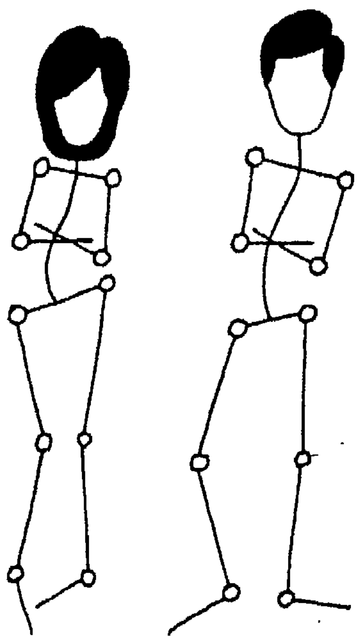
الأكتاف والأرداف المتعارضة :

يمكننا (Pose) شخصيتنا بحيوية أكثر إذا نحن أمسكنا أساس معارضة الأكتاف والأرداف كما هو مبين في الأمثلة شكل (٤ ـ ١٠) فمثلاً، إذا رسمنا الأكتاف مائلة لأسفل نحو اليمين، حينئذ يجب أن نرسم الأرداف مائلة لأعلى نحو اليمين . وهذه فكرة هامة تستعمل بكثرة للموضحين المحترفين ، ورسامى الكارتون ورسامى الرسوم المتحركة . ويمكننا تحسين الاثر الديناميكى لرسومات الشخصيات المتحركة باستعمال أكتاف متعارضة وأرداف كلما كان ممكناً . استعمال أصابع (Stick) مفيد أيضاً لدراسة الحركة المفصلية . الإنسان الأوتوماتيكى (androild) المبين فى شكل (٤ ـ ١١) يبين بعض العقد المستعمل للرسوم المتحركة بمساعدة الحاسب الآلى لحركة مفصلة . هذه العقد مفيدة عندما نرسم شخصيتنا ، ولكنها

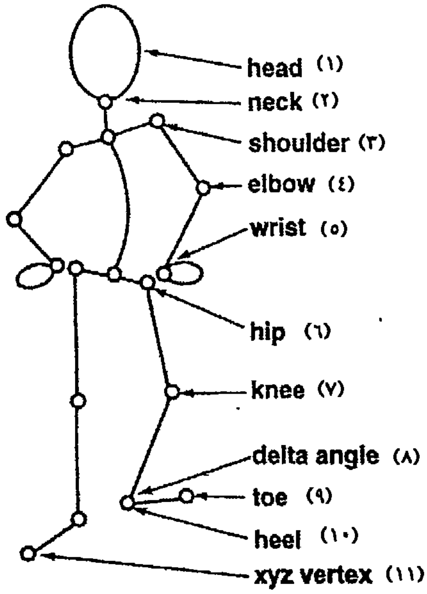
تعطى أيضاً نقط فعالية لبرنامج والذى يحسب أوتوماتيكياً موقع الركبة ، مثلاً ، مؤسس على موقع عقدة الردف. استعمال هذا الاعتماد بين العقد (nodes) يسمى الرسوم المتحركة للحركة المفصلية (articutated) اليد المبينة فى شكل (٤ ـ ١٢) توضح بعض العقد والزوايا المفيدة للرسوم المتحركة المدعمة بالحاسب الآلى لحركة ذات مفاصل .

مقياس رسم أجسام الشخصيات :

الشخصيات المبينة فى شكل (٤ ـ ١٣) مرسومة بطول أربعة رؤوس وهذا مقياس عيارى الصناعة . ورسم حى للآدميين بحجم ستة أو خمسة رؤوس فى الطول . أداء كتاب هزلى لبطل فائق عادة بمقياس بطول ثمانية أو تسعة رؤوس . شخصيات الكارتون سواء كانت فى نسق كتاب هزلى (comic) أو فيلم رسوم متحركة ترسم عادة بطول ثلاثة أو أربعة رؤوس .

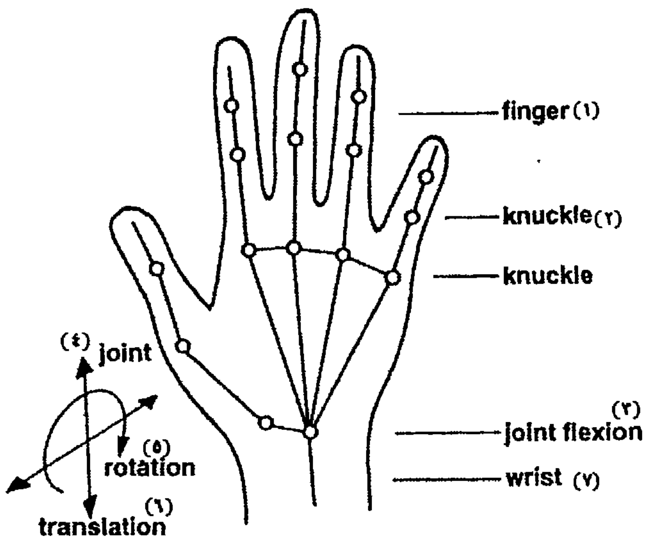


شكل (٤ـ١٠) : يمكن عمل وقفة واقعية بتهيئة زوايا الأكتاف والأرداف لزوايا متعارضة



شكل (٤-١١) : عقد الهيكل العظمى والتى يمكن استعمالها لحركة مفصلية (articulated) لعمل الرسوم المتحركة للانسان الميكانيكى .

(١) الرأس .	(٢) الرقبة .
(٣) الكتف	(٤) الكوع
(٥) المعصم	(٦) الردف
(٧) الركبة	(٨) زوايا داليه
(٩) إصبع االقدم	(١٠) الكعب
(١١) محور xyz	



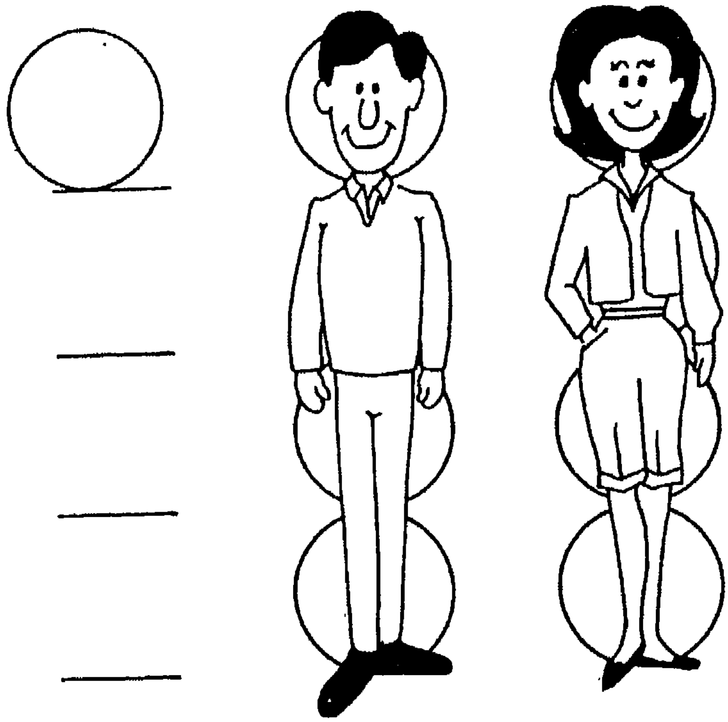
شكل (٤-١٢) : هيكل عظمى لليد البشرية والذى يمكن استعماله للرسوم المتحركة لحركة مفصلية لانسان ميكانيكى

(١) إصبع . (٢) عقله . (٣) مفصل (٤) وصلة . (٥) دوران (٦) انتقال . (٧) معصم

رسوم العيون المتحركة :

عيون الشخصية غالباً هى أهم العناصر أهمية ، خاصة فى الرسوم المتحركة . دراسات متعددة بواسطة وكالات الاعلان قد وضحت هذه المواضيع وقضت وقت أكثر للنظر للعيون لصورة وجه شخص (portrait) عن أى جزء آخر فى الوجه فى الصورة .

نوعان من حركة العين هامتان لرسوم الحاسب الآلى المتحركة للشخصيات، وهى تمسح (Sweeps) العين وتومض (blinks)، وكلاهما سهل عمله ، ولكن من الصعب عمله جيداً .



شكل (٤-١٣) : مقياس رسم صحيح يمكن الوصول له لرسم شخصيات الكارتون أربعة رؤوس .

نظرة (gaze) المسح المحدقة (النظرة الفاحصة الماسحة) :

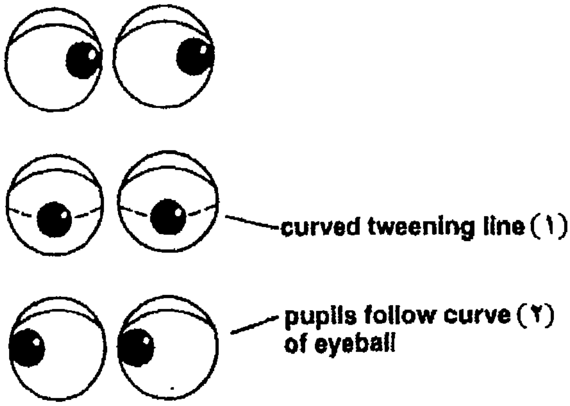
حركة العين المبنة فى شكل (٤ ـ ١٤) توضح الطريقة الصحيحة لرسم نظرة ماسحة متحركة ، وكما يبين خط البين المنحنى ، يجب أن يتبع الطلبة منحنى مقلة العين إذا كانت حركة العين يجب أن تظهر حية. حتى مع ذلك ، لازالت توجد حيلة إضافية يمكننا إضافتها للرسومات المتحركة لنظرتنا الفاحصة الماسحة لجعلها مؤثرة أكثر . إذا لاحظنا الاشخاص الذين حولنا بعناية ، فسنلاحظ أنهم(يتطرفون بعيونهم) (blink) كثيراً عند بدء حركة عين ماسحة . ويعتقد الباحثون أن تطرف العين قبل الحركة تنظم بواسطة العقل الغير واعى (Subconscious) ، وتلعب دوراً هاماً فى تزيين العين قبل الحركة . وإذا أضفنا طرفة عين عيارة عند بداية كل مسح عين كبيرة فى رسومنا المتحركة، فنحن سنضيف إجراء كبير من الحرفية فى عملنا .

طرفة العين العيارية: tweening

توليد صور بينية أقحمها البرنامج من إطار مفتاح ومزودة بواسطة المستخدم أو المطور

الفرق بين طرفة العين العيارية (المعتادة) وطرفة عين نائمة مبين فى شكل (٤ ـ ١٥). فى طرفة العين العيارية ، فإن الجفن العلوى يمر فوق إنسان العين (pupil) والتى تظل هى نفسها ثابتة .

أثناء طرفة العين النائمة ، فإن إنسان العين يسحب لأسفل عند مرور الجفن العلوى عليها .

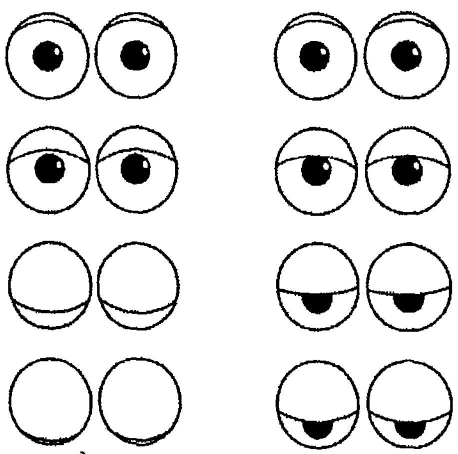


شكل (٤-١٤) : استعمال مسار منحنى لانشاء حركة عين قابلة للتصديق .

(١) خط بين (Tweening) منحنى .

(٢) منحنى متابعة االطلبة لمقلة العين .

المثال فى شكل (٤ ـ ١٥) يبين إطارين بينيين وإطارى مفتاح . كقاعدة عامة ، يجب أن تكون رسومات الطرفات (blinks) المتحركة سريعة مع قليل من البينيات بقدر الامكان . وكلما زادت البينيات (inbtweens) التى نضيفها للطرفة العيارية ، كلما قل نصوع الشخصية فى مظهرها. الدورة القصوى الموصى بها هى إطار مفتاحين (مفتوح كلية ، ومقفل كلية) ، رسم إنهيار واحد ورسمين بينيين .



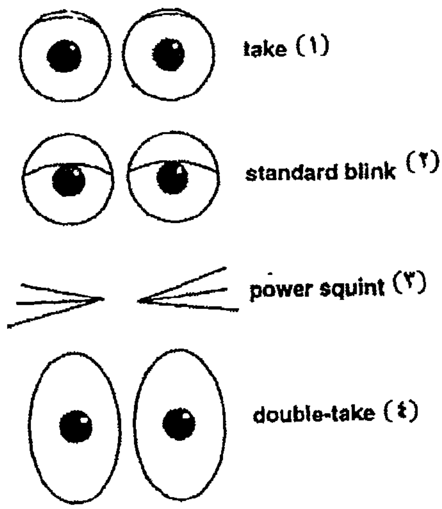
شكل (٤-١٥) : الفرق بين طرفة عين عيارية وطرفة عين نائمة .

(١) طرفة العين العيارية .

(٢) طرفة العين النائمة .

رد الفعل المتأخر :

رد الفعل المتأخر (double take) عبارة عن رد فعل شديد مقدم لبعض الأحداث . الشخصيات تلاحظ الأحداث ، ينظرون بعيداً ، ثم ينظرون مرة أخرى بطرفة عين مثل الموضح فى شكل (٤ ـ ١٦) يجب أن تجعل رسوم متحركة بسرعة، والعنصر النهائى لدهشة العين الواسعة يجب أن يحفظ على الشاشة لفترة أطول .



شكل (٤ ـ ١٦) : أخذ ـ مزدوج بطرفة عين ، يستعمل غالبا ليصور أو يرسم صدمة أو إندهاش.
(١) أخذ
(٢) طرفة عين معتادة
(٣) نظرة شذرة قوية
(٤) رد فعل متأخر

كيف يتم عمل رسوم متحركة للحديث والحوار :

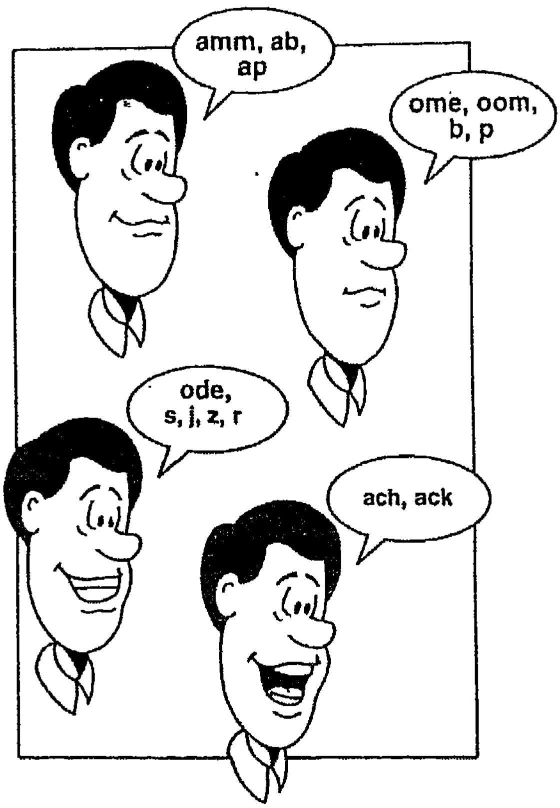
زيادات الوسائط المتعددة لبيئة ويندوز للرسم تجعل من الممكن عمل رسوم متحركة للحديث والحوار، بشرط أن يكون حاسبنا مزود بكارت صوت متوافق / مع الوسائط المتعددة وأدوات تطوير برنامج وسائط متعددة ، ويمكن أن نضيف مسار صوت (Sound track) لرسومات شخصيتنا المتحركة .

ويجب تلبية المعايير حتى يتم جعل الحديث رسوم متحركة . أولاً ، التوقيت لرسوم الشخصية المتحركة ييجب أن يتزامن مع مسار الصوت. شكل فم الشخصية يجب أن يقترح حروف لينة (Vowels) بدقة والحروف الساكنة (Con-sonants) التى قد تم كلامها بواسطة الشخصية (عبر مسار الصوت) .

ستوديوهات الرسوم المتحركة الحرفية عرفت منذ فترةطويلة أشكال الفم الصحيحة لعمل رسوم متحركة للحديث ، ولكنها حرست المعلومات بقوة حتى تحفظ الميزة التنافسية لها فى الأسواق . التوضيحات فى هذا اللباب تعطينا المعلومات الداخلية التى نحتاجها حتى نعمل رسوم متحركة للحديث بنجاح .

الحروف الساكنة (Consonants) :

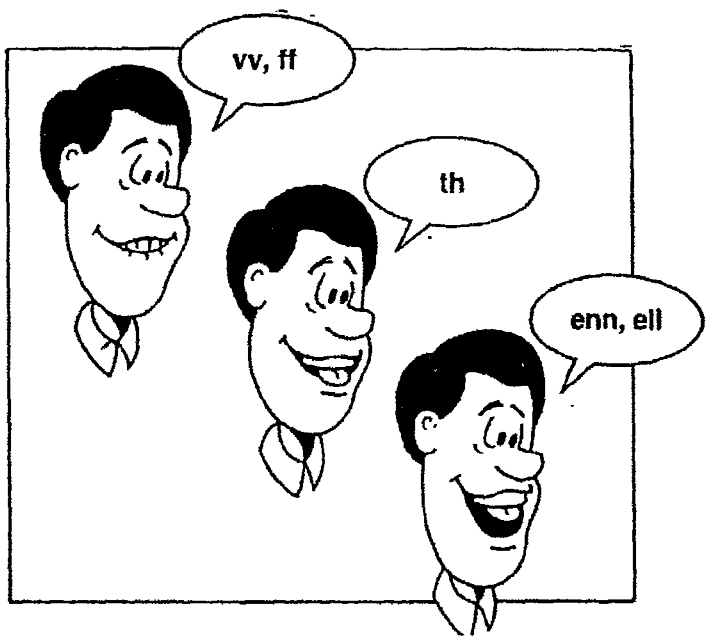
أربعة أمثلة لمجموعات الحروف الساكنة / اللينة التى يتم التكلم بها عند سرعة معتادة والانثناء مبينين فى شكل (٤ ـ ١٧). ومن المهم الفهم أن صوت الحرف الساكن الحقيقى لن يستغرق أكثر من كشفين (عرضين) فى رسوم متحركة لفيلم تقليدى .



شكل (٤ ـ ١٧) : أشكال الفم لمجموعات شائعة للحروف اللينة والساكنة . يتم التكلم بها عند سرعة وانثناء معتادين .

عمل رسوم متحركة بويندوز ، سنرى أن هذا يعنى أن الحروف الساكنة يجب ألا تتجاوز إطار واحد فى تطبيقات رسوم متحركة مؤسسة على ويندوز.

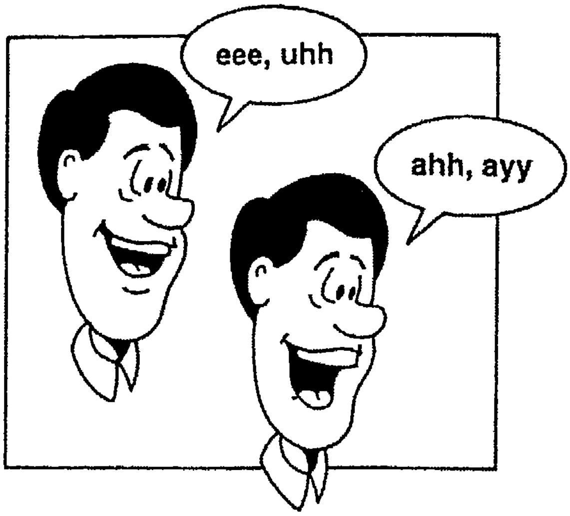
شكل (٤ ـ ١٨) يبين ثلاثة أمثلة لحروف ساكنة تم الكلام بها ببطء (deliberately) . وعند دراستنا لهذه الأمثلة ، يلاحظ أن عناصر كثيرة لفم الشخصية تلعب أدوار هامة بالتساوى فى محاكاة الحديث ، متضمنا شكل الشفاة ، وموضع اللسان وظهور الأسنان، وشكل فتحة الفم .



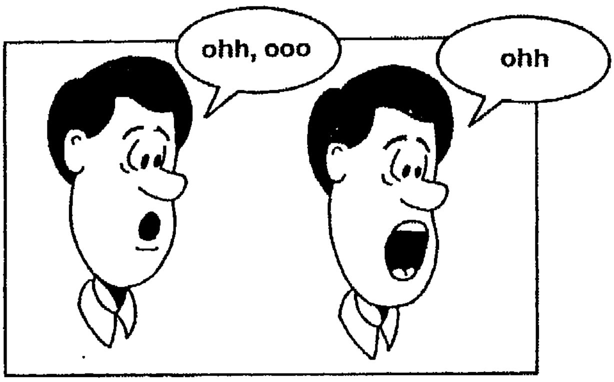
شكل (٤ـ١٨) : أشكال الفم لحروف ساكنة تم التكلم بها ببطء وتأن .

الحروف اللينة (Vowels) :

أمثلة الحروف اللينة التى يتم الكلام بها ببطء وتأنى مبينة فى شكل (٤ ـ ١٩) . أشكال الفم لكلمة oh الشائعة الاستعمال وooh حروف لينة مبينة فى شكل (٤ ـ ٢٠).



شكل (٤ ـ ١٩) : أشكال الفم لحروف لينة (Vowels) تم التكلم بها ببطء وتأن .



شكل (٤ ـ ٢٠) : أشكال الفم للحروف اللينة الشائعة الاستعمال (oh ، ooo)

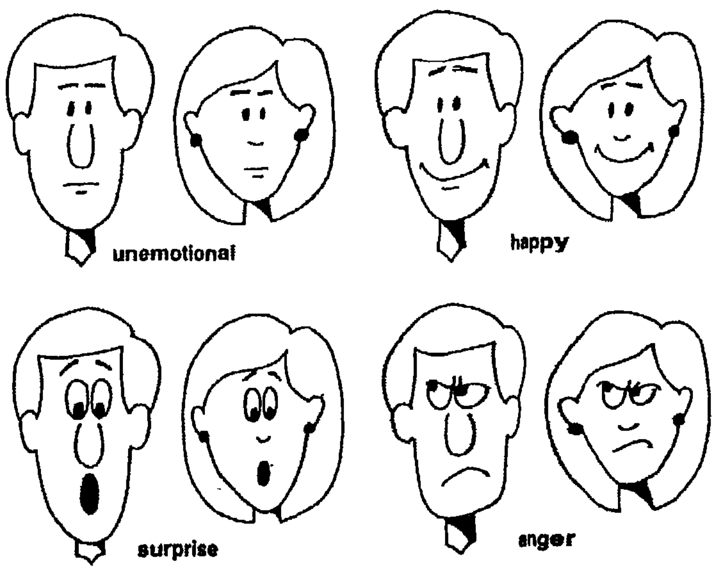
كيفية رسم التعبيرات الوجهية :

أفضل مصدر للعينات للتعبيرات الوجهية للشخصيات ، ولا يهم إذا كانت حقيقية أو كاريكاتيرية أو كارتون ـ هى الحياة ذاتها . الناس الذين نعمل ونلهوا

معهم يمكنهم تقديم مصدر لاينضب من الأفكار. وحتى نكون قادرين على إجابة الضرورة لتعبير وجهى فليس بالضرورة سهلاً كما يبدو . تعبيرات العينات فى هذا الباب تعطينا بيان مفصل يعمل لتعبيرات وجهية والتى يمكننا استعمالها كلما بنينا تتابعات رسومات متحركة لشخصياتنا .

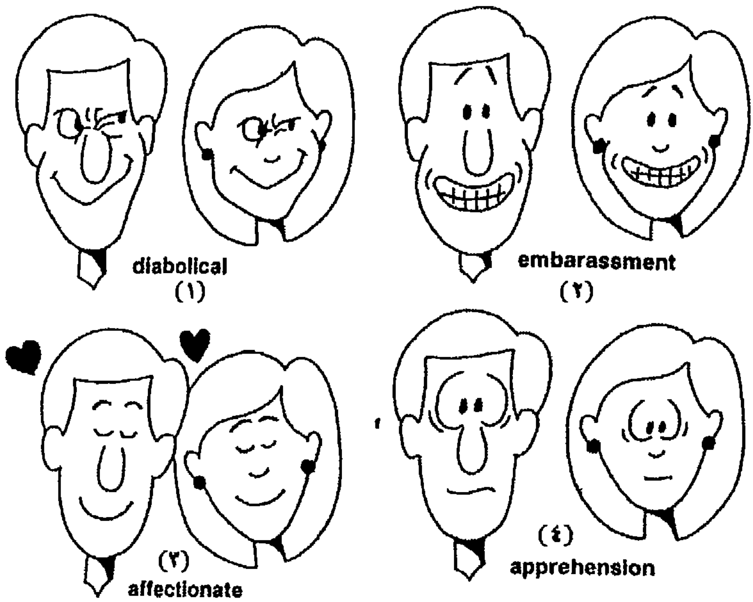
مناظر أمامية لتعبيرات وجهية :

أمثلة لتعبيرات وجهية لمنظر أمامى ذكر وأنثى غير عاطفى (unemotional) وسعيد، ومندهش والغضب مبينة فى شكل (٤ ـ ٢١) .



شكل (٤ ـ ٢١) : منظر أمامى لتعبيرات وجهيه أساسية مستعملة فى شخصيات كارتون للرسوم المتحركة بالحاسب الآلى .

أمثلة لشيطانى ، ومرتبك (embarrassed) ، وحنون (affectionate) وقلق (apprehensive) فى منظر أمامى تم وصفها فى شكل (٤ ـ ٢٢)



شكل (٤-٢٢) : منظر أمامى لتعبيرات وجهيه متخصصة مستعملة فى شخصيات كارتون للرسوم المتحركة بالحاسب الآلى .

(١) شيطانى

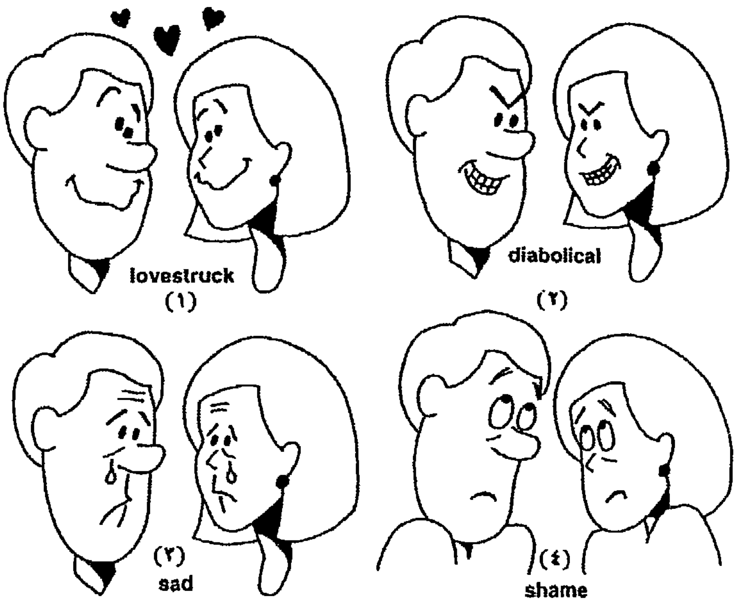
(٢) إرتباك

(٣) جنون / محب

(٤) فهم / إدراك

مناظر تعبير وجهى $\frac{٣}{٤}$:

أمثلة لتعبيرات وجهيه ذكر $\frac{٣}{٤}$ وأنثى فى حب (love struck) ، وشيطانية (diabolical)، والحزن والخجل مبينة فى شكل (٤ ـ ٢٣). أمثلة تعبيرات وجهيه منظر للكريه (distaste ful) ، والخوف ، والقرار (determination) ، والأناقة (smugness) مبينة فى شكل (٤ ـ ٢٤) .



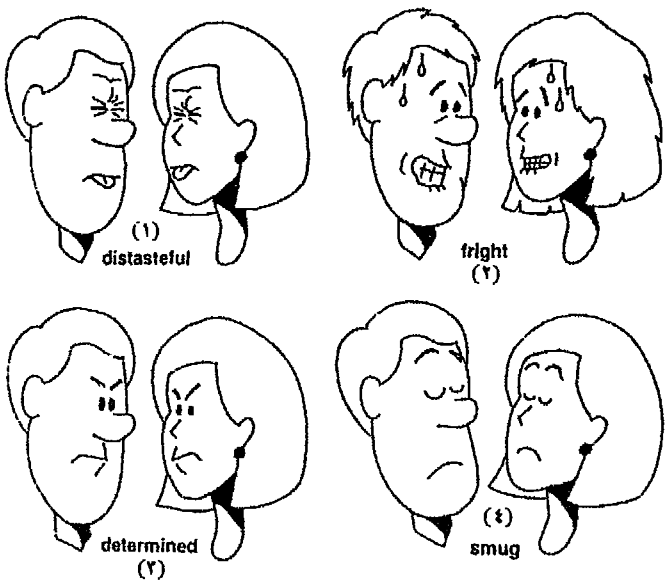
شكل (٤-٢٢) :تعبيرات وجهيه مفيدة فى $\frac{٣}{٤}$ منظر لشخصيات كارتون رسوم متحركة بالحاسب الآلى .
(١) لوعة حب (٢) شيطانى (٣) حزين (٤) خجل أو خزى

وسائل التعبير الوجهى :

بالتعرف على المقومات (ingredients) الضرورية والمتعددة للتعبيرات الوجهية يمكن أن نتعلم بسرعة أن نرسم بأنفسنا من الحياة أو من المرآة . العنصران الأكثر أهمية هما مجموعة العين / الحاجب والفم . ويلاحظ تعبيرات الدهشة والغضب فى شكل (٤ ـ ٢١) مثلاً :العين مفتوحة ـ الحواجب ترتفع دراماتيكيا، والفم يفتح واسعاً فى تعبير دهشة . فى تعبير الغضب فإن العين مفتوحة أيضاً ، ولكن الحواجب تنخفض ، وخط الفم يمتد عبر الوجه فى قوس . كذلك ، ويلاحظ كيف أن خط الفك (jawline) فى هذين التعبيرين يختلف. يمتد الفك لأسفل ليناسب الفم المفتوح فى تعبير دهشة ، بينما الفك أوسع للفم المتجهم (grimacing) لتعبير الغضب التفاصيل حول العينين يمكن أيضاً أن تلعب دوراً هاماً فى التعبيرات

الوجهية . وتلاحظ التفاصيل فى التعبير الشيطانى (diabolical) المبن فى شكل (٤ ـ ٢٢) . وكذلك ، نلاحظ الخطوط حول العينين فى تعبير القلق . مع ذلك ، فإـ الفم يؤدى كل العمل فى تعبير إرتباك فى شكل (٤ ـ ٢٢) . مع أن الارتفاع والموضع للحواجب يعتبر عامل أيضاً .

موضع واتجاه الرأس هام أيضاً كما هو مو ضح بتعبر الخجل فى شكل (٤ ـ ٢٣). العيون الغير مرفوعة والحواجب هما مجرد جزء واحد فقط لكل التعبير، مع الرأس مائلة للأمام ، والذقن تأكل أو تشرب بنهم (tucked in)، هز الأكتاف استهجاناً (shrugged)، هذا على العكس مع التعبير الأنيق فى شكل (٤ ـ ٢٤) ، حيث تلقى الرأس للخلف والفك متهجم thrust للأمام والرقبة مشدودة للخارج. موضع وأشكال العينين والحواجب والفم ليست كافية لاقتراح أناقة (smugness)، واتجاه الرأس ذاتها هى عامل مفتاح .

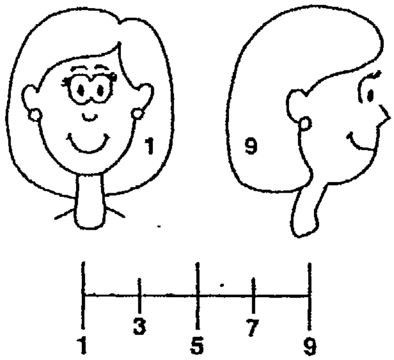


شكل (٤ ـ ٢٤) :تعبيرات وجهيه متخصصة مناسبة لشخصيات كارتون رسوم متحركة بالحاسب الآلى مبينة فى $\frac{٣}{٤}$ منظر.

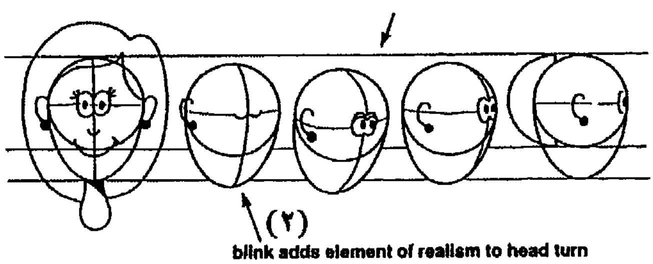
(١) كريه (٢) رعب (٣) مصمم (٤) أنيق

كيفية عمل رسوم متحركة لرأس تدور :

واحد من أهم حركات الرأس الشائعة برسوم متحركة هو الدوران . رسومات مفتاح العينة والرسومات البيانية البينية لدوران رأس مبينة فى شكل (٤ ـ ٢٥) . ومثل حركات العين ، فإن الرأس الدوارة سهلة فى الرسومات المتحركة ، ولكن صعبة فى الرسومات المتحركة الجيدة . المثال المبين فى شكل (٤ ـ ٢٦) يبين كيف تستعمل حركة العين وخط المنحنى بينى (tweening) لجعل الرأس تدور وموثوق بها (authentic) أكثر. ويمكن استعمال ثلاثة حيل لجعل رأس بالرسوم المتحركة تدور أكثر حرفية. أولاً ، جعل رأس الشخصية يمثل قليلاً للأمام وتتبع مسار منحنى أثناء الدوران ، كما هو مبين فى شكل (٤ ـ ٢٦) . ثانياً نتأكد أن الشخصية تنظر بعينين نصف مفتوحين (blinks) عند بداية دوران الرأس .



شكل (٤-٢٥) :رسومات مفتاح ورسم بيانى بينى لدوران رأس



شكل (٤-٢٦) :كيفية استعمال منحنى لعمل الأوضاع للبينيات لعمل دوران رأس قابلة للتصديق .

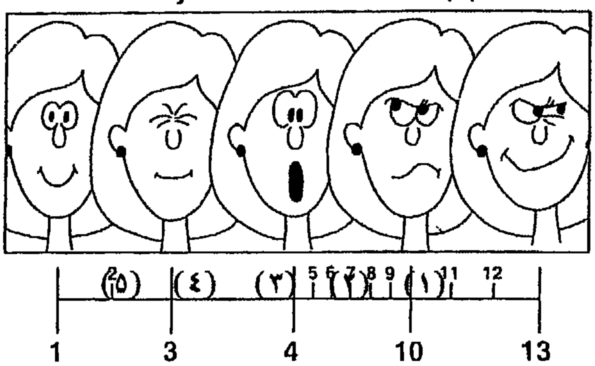
(١) العينان تقودان حركة دوران الرأس .

(٢) نظرة تضيف عنصر الواقعية لدوران الرأس .

التحضير لتطبيق العينة :

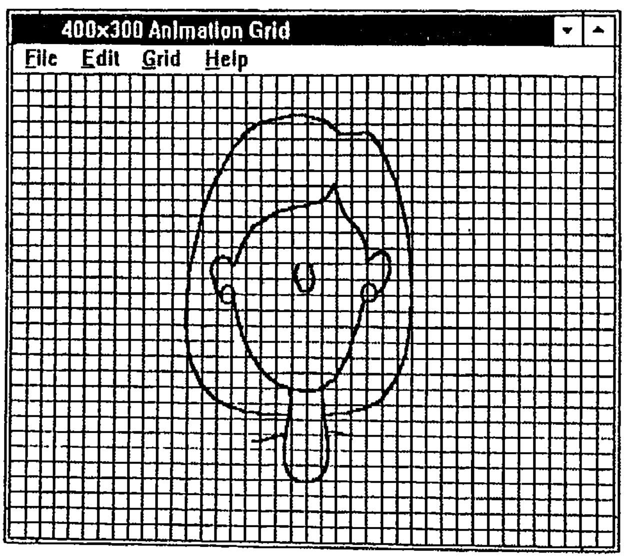
البينيات (inbetweens) : أثناء التحضير لرسوم العينة المتحركة المقدم بعد ذلك فى هذا الباب ، مجموعة من رسومات المفتاح ورسوم بيانية بينية استنتجت سواء قام المؤلف باتخاذ القرار الأفضل عن العنصر الوجهى الذى سيحفظ ثابتاً مثل التتبعات الخلفية) وأى منها يتم برسوم متحركة .

Keyframes and tweens (١)



شكل (٤ ـ ٢٧) :رسومات مفتاح بينية تستعمل لانشاء تطبيق العينة فى هذا الباب .

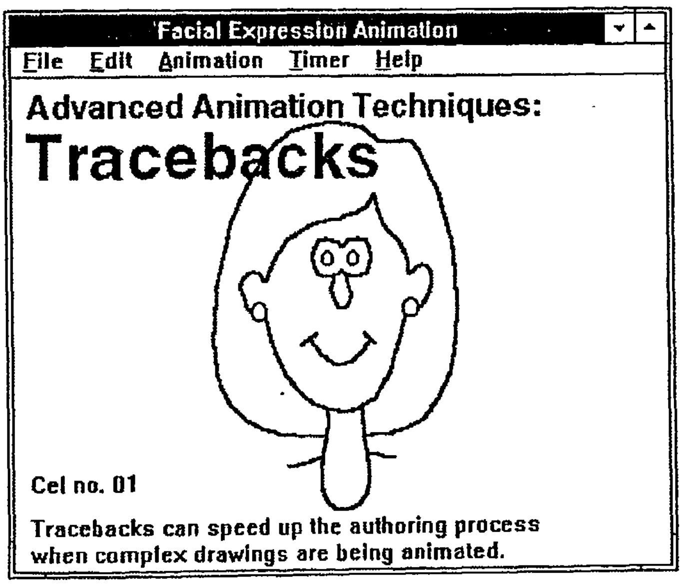
(١) إطارات كبيرة وصور بينية .



شكل (٤ ـ ٢٨) :جزء التتبع العكسى لـCels، موضوع على لوحة الشبكة والذى كان مستعملاً لاستخراج، إحداثيات عرض xy .

تطبيق عينة :

صورة الشاشة فى شكل (٤ ـ ٢٩) تبين الإطار الأول من تطبيق العينة blinx. exe . هذا البرنامج يوضح رسوم وجهية متحركة ، متضمنة رد فعل متأخر (double take) . معدل التفاعل البينى (interactive editor) يسمح لنا بالتجربة مع خيارات إعادة عرض رسوم متحركة مختلفة ، متضمنة ، للأمام كلية ، وللعكس كلية ، وتوقف (pause) ، وإطار متجمد (Freeze Frame) ، وخطوة واحدة للأمام ، وعكس خطوة واحدة وتستمر الرسوم المتحركة فى العرض حتى إذا كانت ويندو مغطاة جزئياً بواسطة ويندو أخرى. أو إذا كانت ويندو مقصوصة جزئياً بطرف الشاشة blink. exe demo ، تبنى من أربعة وحدات، متضمنة وحدة شاملة blink. bas ووحدة شكل blink . Frm ، ووحدة بدء blik 01. bas ووحدة وظيفية blink 02 . bas . كل ملفات الانتاج هذه قد تم تقديمها كشفرة مصدر مستعد ـ لا ـ استعمال فى قائمة البرنامج ، فى هذا الباب ملفات المصدر تزود، أيضاً على القرص المصاحب .



شكل (٤ ـ ٢٩) :إطار تجمد من تطبيق عينة الرسوم المتحركة . نقارن (rendering) هذا الحاسب مع رسوم مفتاح رسم / يد .

ماذا يفعل البرنامج : مرشد المستخدم :

تطبيق العينة يعطى برنامج متفاعل بينى.(interactive platform) للتجربة مع رسوم إطار متحركة لتعبيرات وجهية للكارتون . ويمكن أن نستعمل إما فأرة أو لوحة المفاتيح للتجربة مع برنامج demo. كذلك ، فإن القوائم تدعم مفاتيح خاصة بالذاكرة (mnemonic) . لمشاهدة رسالة مساعدة زمن تشغيل ، ننتقى Help من قائمة Help أو نضغط F1.

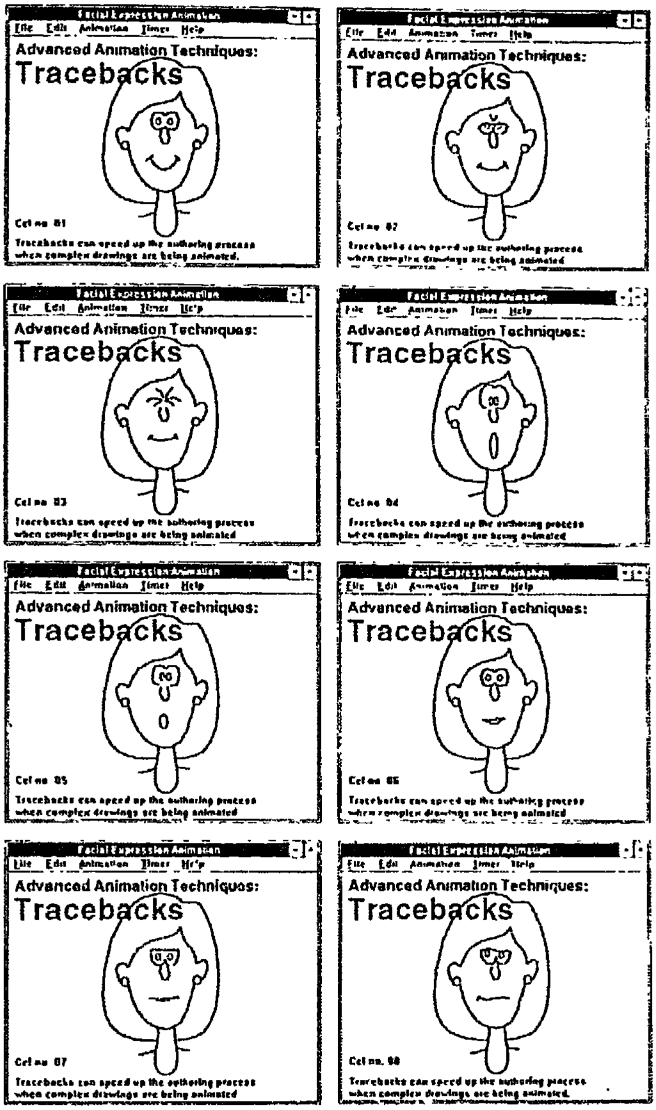
إضافة إضاءة وظلال ولون لمنظر ذو ثلاثة أبعاد : rendering

بدء النظام :

ننتقى بدء النظام (initialze system) من قائمة Animation (رسوم متحركة لبدء الاطار المختفى . ولا يوجد أى من خواص التأليف أو العرض لـ blink. exe ستعمل حتى يتم إنشاء الإطار المختفى .

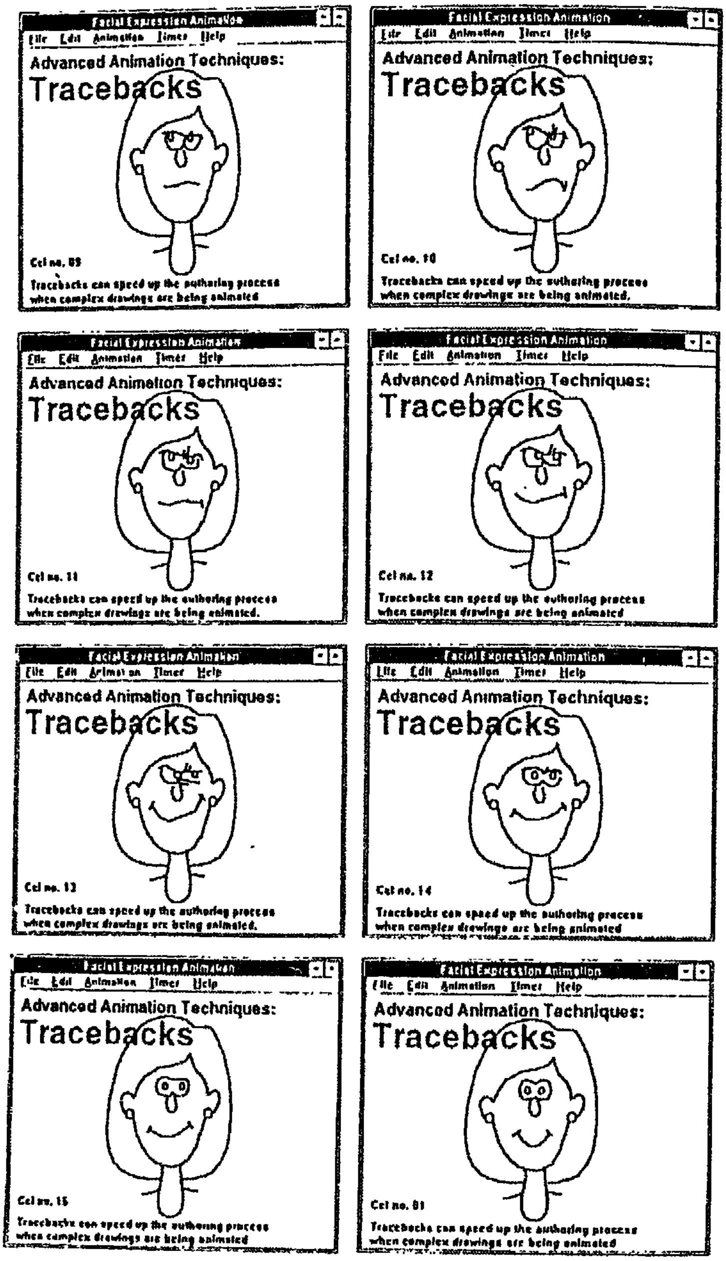
إنشاء الإطارات :

ننتقى Create Frames من قائمة Animation ، لإنشاء تتابع الرسوم المتحركة وتختزنها على قرص . الاطارات الخمسة عشر التى تتضمن الرسوم المتحركة موضحة فى شكل (٤ ـ ٣٠) وشكل (٤ ـ ٣١) . تطبيق العينة سيبنى كل إطار بالمقابل على الصفحة المختفية ، ننسخ الاطار المكمل لويندو العرض ، وحينئذ نحفظ الصورة فى قرص كملف صورة ثنائية. إذا كنا قد حفظنا الخمسة عشر إطار فى قرص ، فإن برنامج demo سيقدم رسالة تحذير وسيعطينا خيار لتجاوز الملفات الموجود بملفات الصورة الجديدة.



شكل (٤-٣٠) :الاطارات الثمانية الأولى لحلقة الرسوم المتحركة الناتجة بواسطة تطبيق العينة عند زمن التشغيل .

(١) وسائل التطبيق المتقدمة .
(٢) التتبعات الخلفية .



شكل (٤-٣١): الاطارات الثمانية الأخيرة لحلقة الرسوم المتحركة.

(١) يمكن للتبعات الخلفية الإسراع في عملية(authoring) عندما يتم عمل رسوم متحركة للرسوم المعقدة.

إذا اخترنا عدم الكتابة على (overwrite) الملفات الموجودة ، فإن تتابع البناء سيتوقف ولا تشيد إطارات أخرى.

إذا لم يتواجد مكان خالى على قرصنا الصلب لاختزان ملفات الصورة الجديدة ، برنامج demo يعرض رسالة استشارية (advisory) ويلغى عملية البناء . البرنامج يبلغ تكملة ناجحة لوظيفة بناء الإطارات (Create Frames) بصندوق رسالة .

تحميل الرسوم المتحركة :

قبل أن نتمكن من لعب تتابع الرسوم المتحركة التى حفظناها فى قرص ، يجب أن نحملها. وإذا حفظنا تتابع الرسوم المتحركة أثناء الدورة العاملة ، ببساطة ننتقى Load Animation من قائمة Animation . وإذا حفظنا الرسوم المتحركة أثناء دورة عاملة سابقة ، يجب أن ننتقى نظام بدء (initialize system) من قائمة Animation قبل انتقاء Load Animation من قائمة Animation.

إذا كان البرنامج غير قادر على إيجاد ملفات إطار على قرص ، فإنه يعرض رسالة إستشارية (advisory) . وإذا حاولنا عرض رسم متحرك قبل تحميله من قرص أو قبل بدء النظام ، فإن برنامج demo سيعرض رسالة إشارة خفية (a hint) .

وفى الظروف الطبيعية ، فإن تطبيق العينة سيحمل كل ملف إطار داخل الإطار المختفى ومن هناك داخل ذاكرة . وبعد تحميل كل الاطارات الخمسة عشر من قرص ، فإن البرنامج يعرض صندوق رسالة لابلاغ النجاح .

عرض الرسوم المتحركة :

لعرض تتابع الرسوم المتحركة ، ننتقى Run Forward من قائمة Animation. وجه الشخصية يتم عمله بالرسوم المتحركة خلال ابتسامة ، أو أخذ مشترك، ودهشة ، واشمئزاز (disgust)، و contemplation ومؤذى ، وحينئذ يعود لابتسامة. صور زمن تنفيذ من تتابعات الرسوم المتحركة كما نرى فى شكل (٤ ـ ٣٠) ، شكل

(٤ـ٣١) . الشخصية ستكمل عرض تعبيرات وجهية حتى نوقف الرسوم المتحركة .

ضبط سرعة عرض الرسوم المتحركة :

يمكننا زيادة أو انقاص سرعة الرسوم المتحركة بالانتقاء من قائمة Timer (الموقت). عند البدء ، فإن معدل العرض هو 18 إطار فى الثانية (Fps) . إذا انتقينا Fps ، مثلاً ، فإن معدل الرسوم المتحركة على الشاشة يتغير . ويمكننا ضبط سرعة العرض عندما تتحرك الرسوم المتحركة للأمام ، أو العكس أو تتوقف .

أداة توقيت منطقية محفوظة : موقت :	Timer
بواسطة ويندوز لارسال رسالة موقت للتطبيق عند فترة معرفة بالتطبيق.	

الاطار المتجمد و الرسوم المتحركة للخطوط المفردة :

التوقف المؤقت (Pause) :

لايقاف الرسوم المتحركة ، ننتقى Pause من قائمة Animation. لاعادة الرسوم المتحركة للعرض فى طور الأمام، وننتقى Run Forward من قائمة Animation لاعادة عرض الرسوم المتحركة فى طور عكسى ، ننتقى Run Reverse من قائمة Animation.

خطوة واحدة (Single - Step) :

أثناء التوقف المؤقت للرسوم المتحركة ، يمكننا تنشيط خطوة مفردة. نضغط مفتاح السهم الأيمن على وسادة مفاتيح (beypad) الاتجاه لعرض الاطار التالى فى تتابع الرسوم المتحركة ، ونظل ضاغطين مفتاح السهم الأيمن للخطو خلال الرسوم المتحركة كلها . لعمل خطوة مفردة عكسياً ، نضغط مفتاح السهم الأيسر على وسادة مفاتيح الاتجاه لاسترجاع العرض الكلى للرسوم المتحركة ، ننتقى إما Run Forward أو Run Reverse من قائمة الرسوم المتحركة.

الرســوم الدائمـة : (Persistent graphics)

تحريك الويندو :

لاختبار خواص الرسومات الدائمة لتتابع الرسوم المتحركة ، نبدأ عرض الرسوم المتحركة ، ثم نحرك الويندو بالطقطقة على قضيب (Caption bar) ونسحب الويندو لموقع آخر على الشاشة . حتى إذا كانت الويندو مقصوصة بطرف الشاشة ، فإن الرسوم المتحركة تستمر فى العرض .

تغطية الويندو :

لتأكيد خواص العمل المتعدد التعاونى للرسوم المتحركة ، نضع الويندو جزئياً على صندوق مجموعة برنامج ، ثم نطقطق على صندوق مجموعة البرنامج . صندوق مجموعة البرنامج سيتحرك إلى صدر الصورة (Foreground) ، جزئياً يغطى تطبيق العينة ، ولكن الرسوم المتحركة تستمر فى العمل .

استعمال قائمة Help :

يمكننا استعمال قائمة Help لاكتشاف حقائق متعددة عن نظامنا عند زمن التشغيل، متضمنة طور الشاشة الحالى ، وأقصى عدد للألوان المتاحة وطور ذاكرة زمن التشغيل (عيارى أو مدعم) .

فحص دقة الشاشة : (Checking the Screen resolution)

لايجاد طور الشاشة الحالى ، ننتقى Screen resolution من قائمة Help . صندوق رسالة سيعرض الدقة الأفقية ـ فى ـ الدقة الرأسية . تطبيق عينةblink. exe يدعم 640x480 ، 640 x 350 ، 600 x 600 ، و 1024 x 768 عروض دقة resolution displays .

أقصى ألوان معروضة :

لتحديد أقصى ألوان متوفرة يمكن عرضها فى نفس اللحظة ، ننتقى Available Colors من قائمة Help. صندوق رسالة سيبين الطور (mode) .Blink. exe يدعم عروض16 لون ، و لونين و 256 لون .

تحديد طور ذاكرة زمن التشغيل :

لتحديد الطور الحالى لذاكرة زمن التشغيل ويندوز ، ننتقى طور Memory mode من قائمة Help . سيظهر صندوق رسالة ، يحتوى على ملاحظة إستشارية تصف الطور وقدراته. برنامج The blink. exe demo يدعم أطوار ذاكرة عيارية ومدعمة.

كيف تعمل شفرة المصدر : مرشد مبرمج :

برنامج توضيح blink . exe عبارة عن تنفيذ مخصص لآلة الرسوم المتحركة إطار النموذج الأصلى . آلات الرسوم المتحركة للاطار. استعمال الوسائل المقدمة فى قوائم البرنامج لـ blink . exe يمكننا أن نضيف بسهولة قدرات رسوم متحركة تعبير وجهى لتطبيقاتنا البيسيك المرئية .

هيئة الوحدات تتبع النسق المنشأ فى تطبيقات للعينة المبكرة .

كيف تعمل الوحدات Form :

وحدات Form تلتصق كثيراً مع النسق المقدم فى قائمة البرنامج لاطار النموذج الأصلى (prototype) لآلة الرسوم المتحركة . آلات رسوم الاطار المتحركة . الكلام الذى يتبع مخصص لعنونة التعديلات التى تمت لشفرة النموذج الأصلى فقط .

الباب الخامس

عمل الرسوم المتحركة في ويندوز

(Animating in Windows)

الباب الخامس
عمل الرسوم المتحركة فى ويندوز
(Animating in Windows)

بيئة رسومات وييندوز لميكروسوفت تقدم برنامج «Platform» للبرمجة ، وهو غنى بإمكانية الرسوم المتحركة . مجال الحاسب الآلى لمساعدة الرسوم المتحركة يجزى وكذلك مربح. ويتمدد بسرعة كبيرة . بيسيك المرئية تقدم بيئة برمجة مناسبة لتطوير تطبيقات رسوم متحركة متطورة .

الرسوم المتحركة فى الحاسبات الآلية الشخصية (PCs) تهيأ لنوعين أساسيين . هذه الأنواع هى التنفيذ والتحكم . التنفيذ يهتم بميكانيكية إنشاء خداع الحركة على الشاشة. التحكم يهتم بادارة الأشياء والأحداث التى يتم عمل رسوم متحركة لها . التحكم هو النطاق الذى يربط مادة الموضوع للبرنامج. التنفيذ (implementation) هو القناة (Conduit) التى تربط البرنامج بالأجزاء الصلبة (hardware) .

التنفيـــذ :

ويندوز تقدم طرق كثيرة مختلفة لتنفيذ الرسوم المتحركة . أكثر الطرق الأربعة العملية والمتوفرة فى ويندوزهى :

- رسوم متحركة للحلقة المثبتة .
- رسوم متحركة للحلقة العاطلة عن العمل (idle)

■ رسوم متحركة مناول / الرسالة مؤسسة على موقت

■ رسوم متحركة دعوة / مباشرة مؤسسة على موقت

هذه الطرق المؤسسة على الموقت والتى تنتج تطبيقات حسنة السلوك (well-behav ed) والتى يمكنها التنفيذ بدون التعارض مع تطبيقات ويندوز أخرى والتى قد تكون فى حالة تنفيذ .

تتابع رسوم متحركة يتم إنشائه : real - time animation
ويعرض فى وقت التشغيل .
تتابع رسوم متحركة مخصص للاستجابة لـ أو التفاعل مع الحوادث التى تحدث فعلياً فى وقت التشغيل .

كل من هذه الطرق الأربعة للتنفيذ يمكن بالمقابل أن تتأثر مثل رسوم متحركة الوقت الفعلى (real - time animation) أو مثل رسوم الاطار المتحركة. الرسوم المتحركة للوقت الفعلى تستعمل صور ديناميكية . رسوم الاطار المتحركة تستعمل الصور الساكنة (Static) . أثناء الرسوم المتحركة للوقت الفعلى (والتى تسمى رسوم متحركة مؤسسة على (Cast - based) بواسطة مبرمجين للوسائط المتعددة)، وكل عنصر صورة يتم رسمه بواسطة البرنامج فى نفس الوقت الذى تعرض فيه الرسوم المتحركة على الشاشة . رسوم الإطار المتحركة من الناحية الأخرى تستجلب وتعرض صور والتى قد تم إكمالها سابقاً وتخزن على قرص . سواء كنا نستعمل رسوم متحركة للوقت الفعلى أو رسوم الاطار المتحركة ، فإن كل صورة مستكملة تسمى إطار (a Frame) .

شكل موجة لتحكم الرسوم المتحركة Cast - based animation
حيث يتم عمل رسوم متحركة لمثلين متعددين أمام منظر .

التحكم (Contral) :

إذا كنا نفهم كيف ننفذ رسوم متحركة على حاسب آلى شخصى والذى ينفذ ويندوز، يمكننا تطوير تطبيقات والتى تقدم دعم قوى للأنواع المختلفة الثلاثة لتحكم الرسوم المتحركة . هذه الأنواع الثلاثة للتحكم هى :

● Scripted animation :

المكافىء المساعد بالحاسب الآلى لصورة ملونة للرسوم المتحركة .

● رسوم متحركة (Scripted).

● رسوم متحركة إجرائية (Procedural).

● رسوم متحركة مؤسسة طبيعياً .

كل طريقة تقحم مثال (Paradigm) مختلف للتحكم فى الأشياء والأحداث قد تم تنفيذها برسوم متحركة .

رسوم متحركة بمساعدة الحاسب الآلى :

تطبيق قوة المعالجة للحاسبات الآلية الشخصية الحالية على المجالات المتعددة للرسوم المتحركة هو فتح قناة لموضوع لفرصة لتعبيرات إنشاء ، والأبحاث العلمية ، ولاتصالات التسويق ، والتسلية والتعليم والتدريب .

الرسوم المتحركة بمساعدة الحاسب الآلى متمركزة على النظم الثلاثة للرسوم المتحركة التقليدية ، والمشاهدة (Visualization) والمحاكاة (Simulation) .

الرسوم المتحركة التقليدية :

الرسوم المتحركة التقليدية تسمى رسوم الفيلم المتحركة أو رسوم Cel المتحركة . فهى تكنولوجيا مدروسة (mature) والتى استنبطت عما يسمى العاب قاعات (parlor) الفانوس السحرى لـ ١٨٢٠ الرسوم المتحركة التقليدية تستعمل التصوير

للإطار المفرد للشرائح "Celluloid" المفردة التقليدية مرصوصة على قمة بعضها البعض ومضاءة من أسفل . وفى كل Cel صورة مرسومة يدوياً لشخصية ، وسناد (a prop) أو خلفية من منظر ، ثم عمل رسوم متحركة له عندما تكون الحاسبات عليها أمر (enjoined) للمساعدة فى انتاج رسوم متحركة تقليدية . وتسمى CATA (أى رسوم متحركة تقليدية بمساعدة الحاسب الآلى -fraditiona Compuler as . (sisted animation :

❀ استعمال الرسومات للبيانات (Visualization) :

استعمال الرسومات للبيانات فى الحاسب الآلى عبارة عن تفكير مرئى بالحاسبات الآلية، يتضمن تلك المجالات المتعددة ، مثل إنشاء شىء ذو ثلاثة أبعاد(3D modeling) ، وإضافة اللون والإضاءة ، وتحليل الصورة ، ورؤية الحاسب الآلى ، والمعالجة الهندسية وتحليل اللون وإعادة اللمس والمقارنات قبل - و - بعد مقارنة محتوى صورتين وأكثر كثير. استعمال الرسومات للبيانات فى الحاسب تدعم قدراتنا على حل المسائل المعقدة والتحقق من الأفكار الخلاقة ، فهو إمتداد لأدوات تفكير عقولنا . إستعمال الرسومات للبيانات يمكن أن يساعدنا على إدراكه وفهمه ، خاصة عندما تكون الصورة برسوم متحركة .

استعمال الرسوم للترجمة : أو المعالجة أو إنشاء بيانات	Visualization

إضافة اللون : rendering والظل والإضاءة لمنظر ذو ثلاثة أبعاد .

الرسوم المتحركة الظاهرة للعيان تسمح لنا برؤية كيف تركب الأجزاء مع بعضها أو كيف يؤثر توقيت الأحداث على بعضها البعض .

الظهور للعيان بالحاسب الآلى هو أداة تفكير هامة جديدة والتى تم جعلها ممكنة بقدرات الرسومات للحاسب الآلى الشخصى . والرسوم المتحركة يمكنها جعل

استعمال الرسوم المتحركة للبيانات صندوق حدود (stand - in) يعتمد عليه للتجارب المعملية . أو النموذج الأصلى الطبيعى أو نموذج بالحجم الحقيقى .

وظيفة معالجة صورة حيث يمكن مقارنة محتوى صورتين	**ensemble processing**

❀ المحاكاة : (Simulation)

استعمال الرسوم للبيانات فى الرسوم المتحركة غالباً لشىء ما أو مما حدث والذى يتواجد فى الحياة الفعلية . مشاهدة شىء بالعين أو حدث بواسطة رسوم الحاسب الآلى المتحركة يتضمن غالباً قاعدة بيانات وقواعد للطباع التى تتخذ من الحياة . فى كثير من الحالات ، فإن الرسوم المتحركة والمحاكاة هى جزء لنفس التقديم .

❀ زمن ـ الفراغ 4D :

رسوم الحاسب المتحركة هى العرض المرئى لزمن ـ فراغ 4D. مفهوم 4D يستنتج من 3D . الأشياء فى الحياة الفعلية تملك الأبعاد الثلاثة للعرض والارتفاع والعمق . هذا هو 3D. موديل3D أو ذو الأبعاد الثلاثة . وعند عرض هذا الشىء بواسطة حاسب آلى ، فإن الصورة تقدم فى فضاء3D باستعمال الاحداثيات X - Y - Z . وعند عمل رسوم متحركة لنفس الشىء ، يضاف البعد الرابع (الزمن) . فالرسوم المتحركة هى حركة عبر الزمن . الشىء المرسوم بالرسوم المتحركة هو عرض باحداثيات 4D زمن ـ فضاء:X - Y - Z - T وحتى يتم إدارة زمن ـ فضاء4D (Space - Time) ، فإن تطبيق رسوم ويندوز المتحركة يجب أن تراقب وتحدث بصفة مستمرة ثلاثة مجموعات الدنياميكيات الرسوم المتحركة . وهذه الديناميكيات هى :

- ديناميكيات الحركة
- ديناميكيات التحديث (update)
- ديناميكيات المشاهدة (viewing)

وتؤخذ مع بعضها ، فإن هذه المجاميع الثلاثة للديناميكيات تصف كل مظهر لتتابع رسم متحرك، ولا يهم إذا كان العرض مقدم مثل تخيل(imagery) 2D أو3D.

❀ ديناميكيات الحركة :

ديناميكيات حركة الرسوم المتحركة ترجع للمرجع والاتجاه ووضع شيء بجانب آخر (juxtaposition) لأشياء منفردة . ديناميكيات الحركة تسمى أيضاً موضع - تغيير - الزمن (Time Varying Position) إذا كان المنظر بالظل والألوان والإضاءة فى 3D ، فإن ديناميكيات الحركة توصف بدوران وانتقال شيء .

❀ ديناميكيات التحديث :

ديناميكيات تحديث الرسوم المتحركة تهتم بالتغيرات فى الشكل واللون والنسيج . أثناء تتابع كرة مرتدة مثلاً ، فإن التطبيق الجيد لديناميكيات التحديث سيؤكد أن الكرة تشوه أو تضغط بطريقة مناسبة فى كل مرة تصطدم فيها بالأرض .

❀ ديناميكيات المشاهدة:

ديناميكيات الرسوم المتحركة للمشاهدة (Viewing) تتضمن تغيرات فى الإضاءة، والكاميرا ونقطة المشاهدة . أثناء تتابع رسوم متحركة يمكن للكاميرا أن تتحرك (Zoom) للداخل أو للخارج وتدور يساراً أو يميناً ، أو تميل الأعلى أو لأسفل، أو تنقل بعربة للأمام أو للخلف أو تتبع جانباً (alongside) شيء متحرك . ديناميكيات المشاهدة تهتم نفسها بهذه الأنواع لحركة الكاميرا ، وكذلك مع تغييرات فى الإضاءة والضوء وتأثيرات خاصة مثل بؤرة لينة (Soft - Focus) .

تحكم الرسوم المتحركة :

طرق التحكم فى الرسوم المتحركة تقع فى ثلاثة أنواع، وهى الرسوم المتحركة scripted، والرسوم المتحركة الإجرائية ورسوم متحركة مؤسسة طبيعياً .

الصور المتحركة بمساعدة الحاسب الآلى (Scripted animation) :

الرسوم المتحركة (Scripted) هى إصدار بمساعدة الحاسب الآلى لرسوم متحركة لفيلم تقليدى . الرسوم المتحركة Scripted تحكم وتوجه بواسطة السمى أى إسم آخر (Namesake) ، النص (the Script) . عادة ، فإن الأشياء المنفردة (تسمى actors) تتحركة أمام خلفية. الأشياء قد تتضمن شخصيات وعناصر مناظر، ودعامات (Props) مطورو الوسائط المتعددة يرمزون لهذه الأشياء المتحركة كدعامات (Cast - members) ، والرسوم المتحركة ذاتها تسمى رسوم متحركة (Cast . based) ، خاصة إذا تم إنتاجها باستعمال وسائل برمجة رسوم متحركة الوقت الفعلى . هذا النوع لتحكم الرسوم المتحركة يسمى أيضاً تحكم واضح(ex-plicit) .

شىء متحرك 2D أو 3D فى رسوم : actor متحركة إجرائية والرسوم المتحركة كيانيتيكية كذلك تسمى «cast meimber»

حيل المهنة :

الرسوم المتحركة المنسوخة تسمى أيضاً رسوم الشخصيات المتحركة ، ورسوم متحركة Cel، والرسوم المتحركة التقليدية (traditional) والرسوم المتحركة للفيلم . وعند استعمال برنامج حاسب آلى للمساعدة فى انتاج الرسوم المتحركة المنسوخة، فإن المبرمج يدخل بيئة حيث أن حيل المهنة تم توثيقها جيداً وإنهائها . ومن المحتمل ، أن الحيلة الأكبر للانتاج المستعملة باستوديوهات الرسوم المتحركة التقليدية هى بينية .

صورة بينية والتى استوفاها البرنامج : من إطار كبير يعطيها المستخدم أو المطور	Tween

❀ الاستيفاء : (Interpolation)

عمل صورة بينية (tweening) هو إنشاء رسومات بينية من مجموعة تسمى رسومات المفتاح (key drawings) والتى تم تحضيرها بواسطة رسام كبير(Senior animator) وكل رسم مفتاح يمثل لحظة هامة فى تتابع الرسوم المتحركة رسام صغير (Junior animators) ترسم كل الإطارات البينية والتى يجب أن تظهر بين رسمى المفتاح (تسمى إطارات المفتاح أيضاً) .

عمل صورة بينية (tweening) مستهلك للوقت ، وعمل مجهد ، وهو تماماً نوع روتينى ويناسبه الحاسبات الآلية . وعند استعمال حاسب آلى شخصى لانشاء صور بينية من إطارى مفتاح، فالمعالجة تسمى استيفاء . الاستيفاء بمساعدة الحاسب الآلى يمكن استعماله مع كلا الرسوم المتحركة 2D، 3D . الاستيفاءات يمكن برمجتها لاتباع خط مستقيم (استيفاء مستقيم) ، أو يمكن أن تتبع منحنى (استيفاء (spline) أو استيفاء منحنى .

خطوات حل توليد صور بينية من إطارات مفاتيح تكون ، والمحور يجب أن تتبع منحنى.	Spline interpolation :

❀ قواعد الرسوم المتحركة :

عدد مما يسمى قواعد الرسوم المتحركة قد تم إكتشافها أو اختراعها أثناء تاريخ ٧٠ عاماً للرسوم المتحركة . الأنواع الثلاثة الأكثر أهمية للقواعد هى :

- التشويه : deformation
- ميكانيكيات الكاميرا .

- staging : توجيه تتابع رسوم متحركة فى إطار مفرد أو رقص (باليه خاصة).
- **التشويه : (Deformation)**

التشويه يرجع للمد والضغط على شيء تحرك أثناء الرسوم المتحركة. الكرة المرتدة مثلاً ، يجب أن تضغط (Squashed) فى كل مرة تضرب سطح الأرض ، ويجب أن تتمدد لتحاكى سرعة عندما ترتد بعيداً .

التشويه المهذب (Subtle) مثل هذا يضيف جودة سحرية وحية لأى رسوم متحركة. فهو يبين أن القائم بعمل الرسوم المتحركة أو المبرمج له دراية بفن الرسوم المتحركة .

ميكانيكية الكاميرا :

قواعد ميكانيكية الكاميرا خاصة بحركة نقطة المشاهدة (viewpoint) ، وتسمى أيضاً حركة الكاميرا . فمثلاً ، فكرة slow - in / slow - out تقول أن أى حركة كاميرا يجب أن تبدأ بطيئة جداً ، وبالتدريج تزيد للسرعة الكاملة ،. وحينئذ يجب أن تقل السرعة وتنتهى بطيئة جداً . حركة الكاميرا لرقص الباليه (Choreographed) دخول بطيء / خروج بطيء، هذه هى الطريقة الوحيدة لتجنب الكاميرا المتقلبة (jerky)، تدور وتتحرك (Zoom).

Slow - in / Slow - out : تجنب لحظة بدء غير طبيعية والتوقف الفجائى للأحداث فى تتابعات الرسوم المتحركة للحاسب الآلى.

الاخراج : Staging

قواعد الاخراج : (Staging) خاصة بتوجيه الرسوم المتحركة . تتابع الرسوم المتحركة، يعد كل شيء يمكن أن يتم إخراجه ويوجه مثل تطوير Broad off " " way play قياس الخطو (Pacing) ، وايقاع (cadence) ، حبكة الرواية (plot)،

وإضاءة دراماتيكية ، مداخل شيء متحرك (actor)، والمخارج ، ودعامات الاخراج المسرحى (props) ، والمشهد (Scenery) ، والأزياء (costumes)، كل هذه تقع داخل مجال الإخراج .

رقص الباليه أو توجيه تتابع رسوم متحركة أو إطار فردى :	Staging
رسوم متحركة موجهة لشيء	Procedural animation

رسوم متحركة موجهة (Procedural) :

تسمى رسوم متحركة actor أيضاً ، وهى رسوم متحركة موجهة / شيء. أثناء عرض الرسوم المتحركة ، فإن البرنامج يحضر الإطار التالى بحساب الموضع التالى لكل شيء فى المنظر . الحسابات مؤسسة غالباً على عشوائية ، وقواعد معرفة بالمبرمج للتصرف . هذه القواعد يمكنها وصف ليس فقط سلوك للشيء فى المنظر ذاته ، ولكن السلوك أيضاً بالنسبة لشيء آخر فى المنظر . وعندما يأخذ شيء تلميحاته من شيء آخر، فإن المعالجة تسمى تفاعل بينى إجرائى .

التفاعل البينى الإجرائى (procedural) :

التحكم الاجرائى (procedural) مناسب أفضل للمحاكاة حيث الفعل يعتمد بينياً بين الأشياء فى المنظر. فمثلاً ، إذا كان تطبيقنا لرسوم متحركة خط تجميع سير حامل ، يمكننا استعمال تحكم إجرائى لتأكيد أن كل صندوق (widgets) يتحرك بسير الحامل حتى نهاية خط الانتاج . وروتين متخصص فى شفرتنا يتعقب موقع السير الحامل من إطار واحد للآخر. الروتين يستعمل الموقع المحدث للسير الحامل لاعادة وضع صندوق الـ widgets للاطار التالى ، يعتمد موقع صندوق الـwidgets (شيء فى المنظر) على موقع السير الحامل (شيء آخر فى المنظر) . هذا الاعتماد بين الشيئين يسمى التفاعل البينى الاجرائى (Procedural interaction) . حركة السير

الحامل ذاته محكومة بقواعد إسمية لتصرف معرف بواسطة المبرمج . هذه القواعد الإسمية هى تحكم إجرائى . الشيء ، ودعامات (props)، والخلفية التى تشكل منظر يمكن إعتبارها واحد من اللاعبين فى تتابع الرسوم المتحركة . الرسوم المتحركة الموجهة ، تشارك فى تشابه الرسوم المتحركة (cast - based) وهى وسيلة تستعمل بواسطة مبرمجى الوسائط المتعددة، كما تم الوصف أثناء المناقشة السابقة للصور المتحركة بمساعدة الحاسب (scripted) .

الرسوم المتحركة المؤسسة طبيعياً :

الرسوم المتحركة المؤسسة طبيعياً ، (وتسمى كذلك الرسوم المتحركة مؤسسة على قيود) (constraint) ، تستعمل قانون الفيزياء للتحكم فى حركة الأشياء أثناء عرض الرسوم المتحركة . الحركة فى نظام مؤسس على قيود مشكلة باستعمال (mod-eled constraint) ، وهو حالة قيد أو قوة . نأخذ فى الاعتبار رسوم متحركة لكرة مرتدة للتنس مثلاً . الجاذبية قوة واحدة بين كثيرين والتى تؤثر على الكرة . والكرة ليست مرفوعة لأعلى بالجاذبية، فهى تدفع لأسفل . والقوة تسمى القيد (constraint) . حركة كرة التنس يحدث لها قيود (constraint) بواسطة قوة الجاذبية . وعندما تضرب الكرة سطح طفلى لقاعة التنس، فهى لا تخترق السطح ، ولكن بدلاً من ذلك ترتد ثانية . كرة التنس لا يمكنها المرور خلال السطح . فهى مقيدة لجانب واحد من السطح .

الرسوم المتحركة المؤسسة طبيعياً تختص بالمواضع والسرعات والقوى والكتلة والقيود، والخ . طريقة التحكم فى الرسوم المتحركة تحتوى على أربعة أنواع :

- الحركة للأمام (Forward Kinematics)
- الديناميكية للأمام
- الحركة العكسية

● الديناميكية العكسية

الحركة (Kinematics) تختص بمواضع سرعات شيء ، والديناميكية تختص بقانون الفيزياء، مثل القوة، والخ ، والتى تتحكم فى تلك المواضع والسرعات .

الحركة للأمام :

الحركة للأمام هى عملية حساب نتيجة التطبيق للسرعة أو العجلة لشيء. الحركة للأمام لا تهتم بذاتها بقوى أو كتلة، ولكنها تأخذ فى الاعتبار الحركة نفسها فقط . يمكننا استعمال الحركة للأمام للفحص إذا تصادم شيئان أثناء تتابع رسوم متحركة .

الديناميكيات للأمام :

الديناميكيات للأمام هى عملية حساب نتيجة التطبيق للقوة ، أو الأحمال أو القيود لشيء . الديناميكيات تعمل بقوانين الفيزياء التي تتحكم فى الحركة للأمام . ويمكننا استعمال التحليل الديناميكى لحساب الحركة (Kinematic) لشيء والناتجة من القوى التى تؤثر على الشيء . كذلك ، يمكننا استعمال التحليل الديناميكى لحساب القوى الناتجة من حركة الشيء. ومثل الحركة للأمام ، فإن الديناميكا للأمام يمكنها التأكد إذا تصادم شيئان ، ولكن الديناميكا الأمامية يمكنها أيضاً حساب القوى الناتجة من التصادم.

الحركات العكسية :

الحركات العكسية هى عملية حساب السرعة أو العجلة المطلوبة لتحريك شيء من موضع لآخر خلال فترة ثابتة من الزمن . وبينما الحركة للأمام تحسب تأثير السرعة أو العجلة وتحدد الموضع الناتج للشيء ، فإن الحركة العكسية تستعمل موضع معرف بالمبرمج ويحسب كمية السرعة أو العجلة المطلوبة لتحريك الشيء لذلك الموضع .

الديناميكيات العكسية :

هى عملية حساب القوى أو القيود المطلوبة لتحريك شيء لكتلة محددة من موضع لآخر خلال فترة زمنية محددة . الديناميكيات للأمام تحسب تأثير القوى والقيود والكتلة والأحمال لتحديد الموضع الناتج وتحسب القوى المطلوبة لتحريك الشيء للموضع .

تنفيذ الرسوم المتحركة :

بيئة الرسم لميكروسوفت ويندوز تقدم عدداً من أدوات المستوى المنخفض والتى تجعل من الممكن تنفيذ صور متحركة بمساعدة الحاسب (Scripted animation) ورسوم متحركة مؤسسة طبيعياً مثل رسوم متحركة (Real - Time) .

> تتابع رسوم متحركة يتم إنشائه وعرضه : real - time animation
> فى وقت التشغيل ، أو مخصص للاستجابة ليتفاعل مع الأحداث فى الحقيقة وقت التشغيل .

أو رسوم متحركة إطار أو رسوم متحركة موجهة لشيء (Procedural) . بالإضافة لذلك ، فإن جودة لغة برمجة البيسيك المرئية ونموذج مؤسس على رسالة لويندوز يمكنه أن يكون مستغلاً لتطوير روتينات لادارة عرض الرسوم المتحركة باستعمال عديد من خطوات الحل (algorithms) القوية . خطوات الحل الأربعة الأكثر فائدة هى :

- رسوم متحركة حلقة ـ مثبتة
- رسوم متحركة ـ حلقة (بدون عمل idle)
- رسوم متحركة رسالة مؤسسة موقت

● رسوم متحركة استدعاء / مباشر، مؤسسة / على موقت.

أدوات المستوى المنخفض :

قدرة ويندوز GDI لمعالجة واختزان مصفوفات الرسومات تقدم موارد فعالة (potent armory)) لأدوات مستوى منخفض مفيدة للرسوم المتحركة. والأكثر فعالية من هذه الأدوات خاصة بمصفوفات الرسومات . مصفوفة الرسم (graphic array) عبارة عن قالب ذاكرة يحتوى على معلومات لصورة مستطيلة . هذه الصورة تسمىbitmap أو bitblt . الصور الأكبر والتى بالحجم الكلى للشاشة أو حجم ويندو الكامل تسمى غالباً بالصور bitmaps. الصور الأصغر تسمى غالباً bitblts . بعض المبرمجين والمطورين لا يفرقون بين الاثنين .

استعمال مصفوفة الرسومات فى برنامج رسومات : (blitting)

عملية bitblt لصق على الشاشة تسمى استعمال مصفوفة الرسومات فى برنامج رسومات (blitting) . وعملياً فإن وظيفة GDI's Bit Blt يمكن استدعائها لنسخ (bitblt) من موقع ذاكرة لآخر ، أو من موضع شاشة لموضع شاشة آخر . نقل قالب أرقام ثنائية (blitting) يعطى الأساسات (under pinnings) للرسوم المتحركة الحالية (real - time) .

نقل قالب أرقام ثنائية تنطق :	Bitblt (bit blit)

نسخ الصفحة :

عملية نسخ bitmap بحجم ويندو تسمى نسخ الصفحة (Page Copying) ، أو إطار (Frame) . الصفحة هى عازل فى ذاكرة وظيفة () GDI's Bitblt يمكن دعوتها لنسخ محتويات ويندو العرض لاطار مختفى فى ذاكرة . كذلك يمكنها أيضاً نسخ محتويات الإطار المختفى (bitmap) مرة أخرى للشاشة (ويندو العرض). نسخ الصفحة يعطى الأسس لصور الاطار المتحركة .

حجب بورقة أو بسيلوفان يستعمله فنانو الرسومات والأفلام: Frisket للرسوم المتحركة لحماية جزء من العمل الفنى حتى لا يقع عليه لون غير مرغوب .

قناع نقل قالب أرقام ثنائية ، حجب الرسومات (Mattes & Fris-kets):

قناع (bitblt) (رسومات القوالب) : matte والذى يستعمل بمصاحبة عمليات هيكل GDI الخطى

بالإضافة لـ pasting (نسخ bitblt أو bitmap من موقع لآخر). يمكن لـGDI أن تستعمل ما يسمى عوامل الهيكل الخطى (raster operators) لعمل تأثيرات مرئية مختلفة عند كتابة bitmap أو bitblt عند الموقع الجديد . عوامل الهيكل الخطى هذه تستعمل المنطق البولى (Boolean logic). وبالتطبيق الحريص بعوامل XOR ، OR يمكن لـ PUT شفافة أن يتم انجازها . عامل PUT الشفاف يستعمل قناع رسومات القوالب (matte) بيضاء لتحضير المنطقة حيث يتم لصق (paste) الصورة. فهى يستعمل حجب (Frisket) لحماية المنظر الموجود من أن يتم باتلافه (overwritten) بجزء الخلفية لمستطيل bitblt . ولأن جزء الخلفية للمستطيل لا يبين فإنه يسمى شفاف . خطوات حل PUT الشفافة تزودنا بقدرة معالجة أشياء متعددة ذات ألوان متعددة عن خلفية متعددة الألوان .

❀ رسوم متحركة فعلية (Real - time animation) :

وهو تتابع رسوم متحركة والتى تنشأ وتعرض فى وقت التنفيذ ، وهو تتابع رسوم متحركة فعلية يستعمل صور ديناميكية . وكل عنصر صورة يتم رسمه بواسطة البرنامج فى نفس وقت حدوث عرض لتتابع الرسوم المتحركة على الشاشة. الرسوم المتحركة للوقت الفعلى يمكن تنفيذها بواسطة ثلاثة نماذج :

- على الشاشة bitblt لـ على الشاشة

- إطار مختفى bitblt إلى على - الشاشة (on - screen)
- إطار مختفى مرسوم إلى على - الشاشة

كل من هذه الطرق لها مميزات وعيوب

على الشاشة bitblt إلى على - الشاشة :

إذا تم نسخ bitblt أو تم تحريكه من موضع على - الشاشة لموضع آخر على الشاشة، فإن ميكانيكيات منع إتلاف الشاشة غالباً تجعل الوسيلة غير عملية . إذا كان bitblt تم تحريكه ، فإن الخلفية عند الوضع السابق يجب أن تسترجع . الارتعاش لا يمكن تجنبه غالباً . سواء تم نسخ bitblt أو تحريكه ، فإن وظيفة شفافة يجب إقحامها عادة لتأكيد أن bitblt ثم لصقه (Pasted) بنظافة عند الموقع الجديد.

الإطارات المختفية bitblt إلى على - الشاشة :

إذا كانت bitblt قد تم إنشائها فعلاً واختزانها فى ذاكرة ، فيمكن استعمالها لبناء صورة كاملة على إطار مختفى . الإطار المختفى (الصفحة المختفية) يتم نسخه لنافذة القرص كإطار تالى فى تتابع الرسوم المتحركة . هذه الطريقة يمكنها عمل الأداء المثالى، ولكنها أحياناً تقيد من قدرة المستخدم على التفاعل المتبادل مع الرسوم المتحركة لأن كل الصور قد تم إنشائها مقدماً . وكلما زاد عدد الصور (bitblts) فإن تطبيقنا له بيان مفصل، وكلما زاد زمن تشغيل تفاعل بينى أكثر والذى يمكننا تقديمه لمستخدمنا .

إطار مختفى مرسوم إلى على - الشاشة :

الطريقة المعددة أكثر لتنفيذ الأجزاء تتضمن رسم كل صورة على إطار مختفى فى وقت التنفيذ (runtime) . وكل صورة مستكملة تنسخ لنافذة العرض أثناء عرض الرسوم المتحركة. هذه الطريقة تعطى أغنى قوة للتفاعل المتبادل للمستخدم مع

الرسوم المتحركة فى وقت التنفيذ ، ولكن عملية رسم كل إطار جديد يمكن أن تكون مضيعة للوقت .

رسوم الإطار المتحركة :

تتابع رسوم الإطار المتحركة (a Frame animation) يستعمل صوراً ساكنة . أثناء تتابع رسوم الإطار المتحركة ، فإن البرنامج يستجلب ويعرض صور نافذة / كاملة والمستكملة سابقاً والتى تم إختزانها على قرص آلة الرسوم المتحركة للاطار المعتادة ، عادة فهى بارعة (smart) بدرجة كافية لمحاولة تحميل كل نتائج الرسوم المتحركة داخل الذاكرة لأداء أفضل للعرض ، إذا سمح الفراغ authoring رسوم الإطار المتحركة والعرض يمكن تنفيذها باستعمال ثلاثة نماذج :

تصميم وإنشاء واختبار تتابع رسوم متحركة	authoring

- bitblt على ـ الشاشة لاختزان الإطار .
- مرسومة على ـ الشاشة لاختزان الاطار .
- عرض إطار لعرض الشاشة .

bitblt على ـ الشاشة لاختزان الإطار :

كثير من المبرمجين يستعملون صور bitblt لبناء كل إطار مستكمل على ـ الشاشة فى نافذة العرض . حينئذ ، فإن البرنامج يحفظ الصورة من على الشاشة فى نافذة العرض. حينئذ ، فإن البرنامج يحفظ الصورة من على الشاشة للقرص الصلب (hard disk) ، إذا كان التطبيق يحتاج أن ينشىء إطارات جديدة فى زمن التشغيل ، يمكن جعل العملية غير مرئية للمستخدم لبناء كل صورة جديدة على صفحة مختفية وحفظ كل إطار مستكمل جديد على القرص الصلب أو الذاكرة .

❑ مرسوم على الشاشة لاختزان الإطار :

حيث أن صور bitblt المنشأة سابقاً ليست عملية ، والمطور سيستعمل غالباً شفرة والتى ترسم كل إطار من المؤقت (scratch) على الشاشة . حينئذ ، فإن البرنامج يحفظ كل إطار مستكمل من الشاشة للقرص الصلب . إذا كانت إطارات جديدة قد أنشئت فى وقت التنفيذ ، فإن العملية يمكن أن يكون محجوباً (con-cealed) من المستخدم بناء كل إطار جديد على صفحة مختفية والحفظ فى قرص صلب أو ذاكرة .

❑ عرض الإطار على عرض الشاشة :

عرض تتابع رسوم إطار متحركة عادةً من قرص لعرض أو من ذاكرة لعرض. الإطارات التى تشكل تتابع الرسوم المتحركة تختزن عادةً على قرص صلب. إذا سمحت الذاكرة ، فإن آلة العرض عادة ستناضل لتحميل كل الإطارات الضرورية داخل ذاكرة، ومن هناك يمكن عرضها بسرعة بنسخها فى تتابع لنافذة العرض على الشاشة . إذا لم تسمح الذاكرة تحميل مسبق ، فإن آلة الرسوم المتحركة مجبرة على أن تحمل كل إطار من قرص عند الطلب أثناء العرض ، وأداء الرسوم المتحرك مربوط مباشرة بأداء قرص صلب .

❀ تشفير تتابع رسوم متحركة :

شفرة البيسيك المرئية والتى تقود تتابع رسوم متحركة يمكنها استخدام خطوات حل حلقة / مثبتة، أو حلقة / لاتعمل (idle)،بدون عمل أو يمكنها الاعتماد على خطوات حل مؤسسة على موقت .

خطوات حل الحلقة :

الحلقة المثبتة :

الحلقة المثبتة (Fixed loop) عبارة عن قالب شفرة والذى ينفذ تكراراً لعدد

محدد مسبقاً وبدون عمل (iterations). كذلك ، فإن الحلقة المثبتة تسمى أيضاً الحلقة اللانهائية بواسطة مبرمجى بيسيك المرئية . الشفرة للحلقة محتواة عادة فى واحدة عن الإجراءات الوظيفية للتطبيق . وبينما الحلقة تنفذ ، فإن تطبيقات أخرى قد تكون فى حالة تنفيذ فى بيئة ويندوز غير قادرة على مشاركة CPU.

شفرة الحلقة المثبتة مفيدة فى التفجرات (bursts) القصيرة للرسوم المتحركة (تسمى رسوم spot المتحركة) ، حيث أن إيجاز (brevity) تتابع الرسوم المتحركة لن يؤثر بضرر على تطبيقات أخرى قد تكون فى حالة تنفيذ .

❀ الحلقة التى لا تعمل : (Idle loop)

تطبيقات بيسيك المرئية التى يتم التصرف فيها جيداً تعدل قوالب حلقتها المثبتة لعمل رسوم متحركة لحلقة لا تعمل . يمكننا عمل ذلك بوضع حلقة Do While ( ) DoEvents فى الأجزاء () Main فى وحدة البدء لتطبيقنا. جزء (Fragment) الشفرة التالية سيعمل مع تطبيقات العينة :

أثناء ما يكون النظام لا يعمل	Do While DoEvents ()
تبين الإطار التالى	z Show Next Frame
تستمر على الحلقات للخلف	Loop

❀ خطوات حل مؤسسة على موقت :

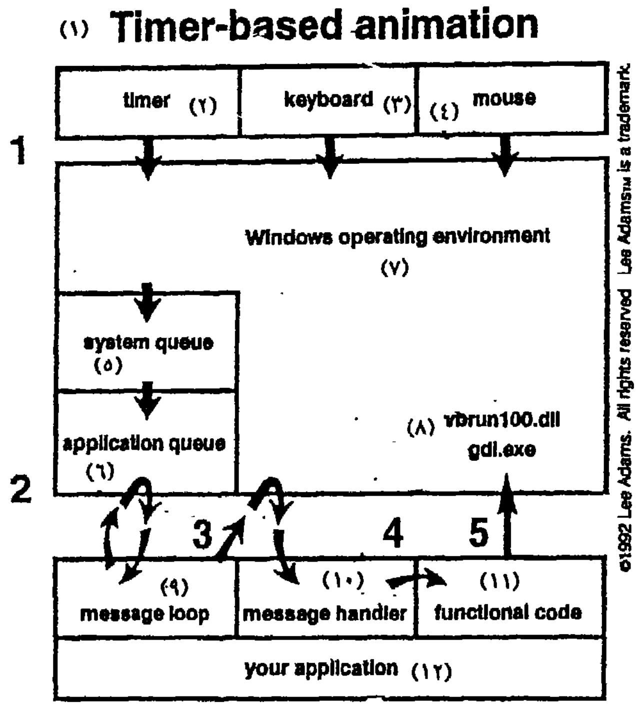
خطوات الحل المؤسسة على موقت (Timer - based algorithms) تقدم أكثر الوسائل الفعالة للتحكم فى عرض تتابع رسوم متحركة . وبعد أن حددنا فترة توقيت بتهيئة خاصية فترة (i nterval) الموقت (والتى تكافىء دعوة لوظيفة GDI,s () Set Timer. ويندوز سترسل دخل موقت لتطبيقنا فى كل فرصة تنقضى فيها الفترة .

نظرياً ، دخل الموقت هذا يمكن أن يأخذ اتجاهين . الأول ، ويندوز يمكن أن ترسل رسالة WM - TIMER لتطبيقنا بنشر (Posting) الرسالة فى الصف ، ومنه يمكن استجلابها بتطبيقنا . هذا الاتجاه يسمى رسوم متحركة مناول ـ رسالة مؤسس على موقت (timer - based message - hanler). وهو النموذج الذى تستعمله بيسيك المرئية . ثانياً ، يمكن لويندوز أن تدعو إجراء محدد فى تطبيقنا فى كل فترة توقيت .

هذه الطريقة تسمى «رسوم متحركة دعوة ـ مباشرة مؤسسة على موقت - Timer based direct - Call animation

رسوم متحركة مناول ـ الرسالة المؤسسة على موقت :

شكل (٥ ـ ١) رسم توضيحى مفهومى والذى يوضح رسوم متحركة مناول رسالة مؤسس على موقت يعمل فى تطبييقات البيسيك المرئية والتى تنفذ فى بيئة متعددة الأعمال متعاونة لويندوز . هذا المثال يعطى أداء زمن تشغيل جيد، ويسمح لنا أن نتحكم بطريقة صحيحة فى سرعة العرض بضبط فترات التوقيت بتهيئة خاصية فترة الموقت ببساطة . طريقة مناول الرسالة المؤسسة على الموقت تعطينا تحكم جيد خلال تطبيقنا خلال زمن التشغيل ، لأن السريان المنطقى يمر خلال مناول الرسالة والتى بنتها بيسيك المرئية فى تطبيقنا ، والتى بالمقابل تدعو إجراء وظيفى لعرض الإطار التالى ـ نموذج مناول الرسالة متعدد الجوانب أيضاً لأنه يقدم طريقتين مختلفتين لعرض الإطار التالى. أولاً، مناول رسالتنا على الخصوص يمكن أن يدعو الوظيفة الصحيحة عند استقبال حدث موقت (Timer) . ثانياً ، أى إجراء فى تطبيقنا يمكن أن يدعو إجراء عرض/ الإطار عندما يحتاج لعمل ذلك ـ هذا موجود لاطار تجمد وتحكمات التفاعل المتبادل خطوة / واحدة .



شكل (٥ ١) : رسم توضيحى مفهومى لرسوم متحركة مؤسسة على موقت فى بيئة ويندوز .ساعة النظام تستعمل لتشغيل الموقت الفعلى لويندوز ، والذى يبعث رسالة فى الصف ، وفيها يسترد بواسطة حلقة رسالة تطبيقنا . حينئذ ، فان مناول الرسالة لتطبيقنا يدعو الوظيفة الصحيحه لتعرض الإطار التالى فى تتابع الرسوم المتحركة . بيسيك المرئية تناول كل الأعمال الروتينة (chores) الأعمال المنزلية الخاصة بادارة حلقة الرسالة ومناول الرساله .

(١) رسوم متحركة مؤسسة على موقت (٢) موقت
(٣) لوحة مفاتيح (٤) فأره (٥) صف النظام
(٦) صف التطبيق (٧) بيئة تشغيل ويندوز
(٩) حلقة الرسالة (١٠) مناول الرسالة
(١١) شفرة وظيفية
(١٢) تطبيقنا

(1) شريحة موقت النظام تستعمل لتحديث موقت فعلى يدار بواسطة ويندوز
(2) ويندوز ترسل رسالة موقت فى الصف لتسترد بواسطة حلقة رسالة تطبيقنا
(3) حلقة الرسالة تسأل ويندوز أن تمرر رسالة إلى وظيفة مناول رسالة تطبيقنا
(4) وظيفة مناول الرسالة تدعو وظائف القلب الصحيحة فى تطبيقنا
(5) وظائف القلب تدعوا وظائف ويندوز Dll الصحيحة

رسوم متحركة دعوة / مباشرة مؤسسة على موقت :

أثناء تتابع رسوم متحركة والتى تستعمل نموذج دعوة/ مباشرة مؤسسة على موقت ، فإن ويندوز ترسل رسائل موقت مباشرة لاجراء محدد فى تطبيقنا . هذا النوع من الرسوم المتحركة يحتاج لمعالجة مباشرة (underpinning) الداخلية لتطبيقنا ، وليس عملياً لتطبيقات بيسيك المرئية. لوصف كيفية كتابة شفرة لرسوم متحركة دعوة / مباشرة مؤسسة على موقت ، ننظر فى أى كتاب آخر عن برمجة ويندوز، عليه عنوان لويندوز.

تهيئات موقت :

خدمة الموقت المزودة بعوامل ويندوز باستعمال شريحة موقت الأجزاء الصلبة ، والتى توزع مقاطعة INT 08H تقريباً 18.2 مرة فى كل ثانية . هذا يعنى أن مدى محدود فقط لتهيئيات الموقت يتم تنفيذه فعلياً بدخل الموقت ، كما فى شكل (٥ ـ ٢).

(١) USABLE SetTimer( ) ARGUMENTS		
(٢) INTERVAL (IN CLOCK TICKS)	(٣) INTERVAL (IN ms)	TIMER EVENTS (٤) PER SECOND
1	55 ms	1000/55 = 18.2 fps
2	110 ms	1000/110 = 9.1 fps
3	165 ms	1000/165 = 6.1 fps
4	220 ms	1000/220 = 4.5 fps
5	275 ms	1000/275 = 3.6 fps
6	330 ms	1000/330 = 3 fps

شكل (٥ - ٢) أداة موقت البيسيك المرئية تعطى وصول مستوى عالى لوظيفة GDI's Set Timer() ويبدأ الموقت بالمنطقة المزدوجه على أداة موقت البيسيك المرئية عند وقت التصميم . وبتغيير خاصية فترة الموقت عند زمن - التشغيل ، يمكننا إرسال خلاصات متعددة إلى set Timer ( ) لتنفيذ رسوم

تابع شكل (٥ ـ ٢)

متحركة compliant ويندوز، محكومة بالموقت مؤسسة على رسالة ويندوز rounds down GDI.خلاصة لأقرب قيمة مستعمله : فمثلاً ، إذا كان تطبيقنا يدعوا، Set Timer () بخلاصة 170 ، فان ويندوز roundsix down إلى 165. شريحة الموقف عند 2 . 18في الثانية ، يعنى كل تكة54.945 ميللى ثانية (m . s) (1000) مقسومة على 2.81) . لذلك فان 170 ms هى 3.094 تكات ساعة (170 مقسومة على54.948)

ويندوز تقرب هذا إلى ٣ تكات لك والتى تكافىء 165 ms

(١) خلاصات USABLE set Time

(٢) الفتراة بعدد تكات ساعه

(٣) الفترة بالميللى ثانيه

(٤) أحداث الموقت فى الثانيه.

فمثلاً ، إذا هيأنا خاصية فترة الموقت إلى 170 ، فنحن نهيىء فترة الموقت إلى 170m.s . هناك 1000 ميللى ثانية فى الثانية الواحدة، ولكن لأن شريحة موقت الأجزاء الصلبة تعمل عند 18.2 تكة ساعة فى الثانية ، وكل تكة ساعة تحدث كل 54.945.m.s (1000 مقسومة على 18.2) . هذا يعنى أن طلب لـ170 m.s فى الحقيقة هو طلب لفترة 3.094 تكات ساعة (170 مقسومة على 54.954) ويندوز تقرب هذا إلى 3 تكات ساعة ، أو 165 m.s ، ومع ذلك ، نحن طلبنا170 m.s فى دعوتنا لـ ( ) Set Timer ، فإن تطبيقنا سيستقبل رسالةWM - TIMER كل165 ms . المقارنة فى شكل (٥ ـ ٢) تبين أيضاً كيف أن دخل الموقت يرجع مباشرة لسرعة الرسوم المتحركة ، والتى يعبر عنها بالإطارات (Frames) فى الثانية (FPS) . ويلاحظ أن دعوة الـ GDI Set . TIMER (170) مماثل للاستعمال عبارة بيسيك المرئية لـ Timer interval = 170 (أى فترة الموقت تساوى 170)

برنامج تصميم وانشاء واختبار تتابع رسوم متحركة Authoring: Plat Form

هذا البرنامج هو البرنامج المستعمل فى نظام حاسب المطور التصميم، وبناء

واختزان تتابع رسوم متحركة . كذلك ، فإن برامج authoring platforms تسمى أيضاً (authoring tools) وأدوات التسجيل.

authoring tools :

أدوات تصميم وانشاء واختبار تتابع رسوم متحركة .

❀ برنامج « Platform » التسليم :

برنامج التسليم Platform هو البرنامج المستعمل للعب تتابع الرسوم المتحركة على حاسب آلى المستخدم. برامج التسليم يسمى آلات العرض ، وآلات الرسوم المتحركة واللاعبين (Players). آلات الرسوم المتحركة الموجودة فى قوائم البرنامج قد تعطى خواص كثيرة موجودة فى لاعبى الرسوم المتحركة للوسائط المتعددة لويندوز، متضمنة تحميل تتابع رسوم متحركة ، ولعبة أمامية ، ولعبة عكسية ، وإطار متجمد ، «وخطوة مفردة ـ أمامية»، «وخطوة مفردة ـ عكسية» وأخرى .

❀ المعالجات الإنشائية :

العمليات الإنشائية خاصة بالإنشاء ، والتعديل(editing) وتطوير بخط القصة الذى سيتم برسوم متحكمة. تتابع رسوم متحركة غالباً سيصمم باستعمال المعالجات الإنشائية التالية :

- نص أو فيلم Script
- لوح القصة Story boar
- (تمثيل مرئى مكتوب لمسار الصوت لتتابع رسوم متحركة) bar Sheet
- صفحة العرض exposure Sheet

وعندما تكون هذه المعالجات الخلاقة كاملة ، فإن عمليات الإنتاج تستعمل لبناء الصور لتتابع الرسوم المتحركة.

❀ عمليات الإنتاج :

عمليات الإنتاج تستعمل لبناء صور تمهيدية (preliminary) للرسوم المتحركة ، إختبار النموذج الأصلى (prototype) للرسوم المتحركة للأداء والتقدم (Pacing) ، ثم بناء الإطارات للرسوم المتحركة المنتهية .

مشروع رسوم متحركة سيعانى عادة لمعالجة الإنتاج التالية :

- لوح قصة بالتفصيل Detailed Storyboard
- إطارات مفاتيح Key Frames
- إطارات بينية Intermediate Frames (tweening)
- رسوم محاولة متحركة Trial animation (Pencil test)
- التحبير والطلاء Inking & Painting

عندما تكون معالجات الانتاج هذه استكملت ، تستعمل معالجات التخزين لحفظ تتابع الرسوم المتحركة للتوزيع والعرض .

❀ معالجات الاختزان :

معالجات الاختزان تستعمل لحفظ الاطارات المستكملة فى القرص الصلب ((hard disk ، أو شريط الفيديو (VTR) ، أو فيلم . معالجات التخزين تختص بما يلى :

- إختزان الصور
- التسليم والعرض
- الرسوم المتحركة لخرج NTSC ، VTR

SetTimer( ) USABLE SETTINGS (١)			(٢) NTSC VIDEOTAPE AND TELEVISION REQUIREMENTS		
clock ticks (٣)	ms	fps (٤)	animated on ... (٥)	ms	fps (٦)
1	55	18.2	twos	67	15
2	110	9.1	threes	100	10
3	165	6.1	fours/fives	133/167	7.5/6
4	220	4.5	sixes	200	5
5	275	3.6			
6	330	3	tens	333	3
COMPUTER PLAYBACK (٧)			(٨) VTR PLAYBACK		

شكل (٥ - ٣) : مقارنة لمعدلات إطارات الرسوم المتحركة بين الرسوم المتحركة للحاسب الآلى الشخصى ورسوم التليفزيون المتحركة . معدل الرسوم المتحركة لـTV، VTRالمفضلة هى30 إطار فى الثانية (fPS) ، مكافىء لمعدل انعاش شاشة NTSC. الصور المتحركة ورسوم الفيلم المتحركة تنتج عادة عند 24 إطار فى الثانية (fps)

(١) التهيئات المستعمله (تهيئة الموقت)
(٢) إحتياجات شريط الفيديو والتليفزيون
(٣) تكات الساعة
(٤) إطار فى الثانية
(٥) توصيل الرسوم المتحركة (aninated on)
(٦) إطار فى الثانية
(٧) عرض الحاسب الآلى
(٨) عرض جهاز تسجيل الفيديو(VTR)

التسليم والعرض لتتابع الرسوم المتحركة يمكن أن يكون من شريط فيديو ، أو قرص صلب أو من جهاز اسقاط أفلام . إذا كان تتابع الرسوم المتحركة مخصص لعرض التليفزيون أو شرط الفيديو ، يجب أن نكون حريصين لاحتياجات التوقيت لشريط فيديو وتليفزيون NTSC كما يبدو فى شكل (٥ ـ ٣) . الرسوم المتحركة للتليفزيون وشريط الفيديو تعمل عند 30 إطار فى الثانية. وهذا سمى عمل رسوم متحركة (animating) على آحاد . وكل إطار لـ $\frac{1}{30}$ من الثانية وظيفةGDI,s Set

Timer ( ) وخاصية Timer. Interval لبسيك المرئية تسمح بأقصى معدل للعرض 18.2 مرة فى الثانية ، كما يظهر فى شكل (٥ ـ ٢) . ولكل الأغراض العملية، فإن هذا المعدل مكافىء لـ animating on twos 15 إطار فى الثانية ، والمماثل لعرض كل إطار لفترة $\frac{2}{30}$ من الثانية .

عرض رسوم متحركة يعرض animating on twos:

كل إطار لفترة زمنية ضعف معدل الإطار . الرسوم المتحركة للتليفزيون وتسجيل الفيديو تتم عادة عند 30 إطار فى الثانية .

إعتيادات الأداء :

رسومات الحاسب الآلى المتحركة تعتمد على المكونات الصلبة وعدد من العوامل والتى خارج تحكم المؤلف يمكنها أن تؤثر مباشرة على أداء تتابع الرسوم المتحركة فى وقت التشغيل . أهم هذه العوامل هى سرعة المعالج (Processor) ، وطور العرض ، وكمية الذاكرة الطبيعية وطور ذاكرة زمن التشغيل .

أطوار ذاكرة ويندوز :

طور ذاكرة زمن التشغيل المستعمل بواسطة ويندوز له نتائج كبيرة لتتابع الرسوم المتحركة . الطور الفعلى (غير مدعم بواسطة ويندوز 3.1 أو بواسطة بيسيك المرئية) يعطى أقصى بيئة مقيدة للرسوم المتحركة ، وستوفر ذاكرة 640K فقط. وآلة الرسوم المتحركة يجب دائماً أن تحمل إطارات من قرص عند الحاجة أثناء عرض الرسوم المتحركة. الطور العيارى (المعتاد) يقدم بيئة محسنة لعرض الرسوم المتحركة، لأنه يتوفر حتى 2MB من الذاكرة والمعالج يعمل فى طور ذو حماية .

الطور المدعم يقدم أحسن بيئة لزمن التشغيل للرسوم المتحركة وحتى 16 MB

(ميجا بايت) للذاكرة الطبيعية متوفر ، وعندما تكون الذاكرة الطبيعية منهكة ، فإن المعالج يقدم ذاكرة واقعية بواسطة بمبالة 4K بلوكات للقرص الصلب . وهذا يعنى أن الذاكرة الممتدة والمتاحة كذاكرة طبيعية على حاسب آلى العرض ، كلما زادت الإطارات التى يمكن تحميلها داخل RAM بواسطة آلة العرض . الشفرة المصممة جيداً ستقدم نتائج مقبولة على كلا برامج التسليم Platforms الطرف العالى والطرف المنخفض .

سرعة المعالج :

سرعة معالج (processor) الحاسب المستخدم لها تأثير مباشر لأداء زمن التشغيل لتتابع الرسوم المتحركة. معدل عرض 15 fps يمكن الوصول له بآلات $25MH_Z$، $33MH_Z$ باستعمال شفرة-لون 16 x 480 x 640 VGA,s . الحاسب الآلى MH_Z سيجد صعوبة للوصول لهذا المعدل ، وظاهرة تعرف «- menu freeze up» قد تحدث أثناء عرض الرسوم المتحركة . ولأن قدرة المعالجة الكبيرة هذه التى تستعمل ببساطة لتحريك الإطار التالى للعرض ويندوز ، قد لا تكون قادرة أن تنتزع (snatch) دورات كافية على المعالج للسماح له بعرض قائمة تم انتقائها بواسطة المستخدم .

المراجع

* Visual Basic .

Animation Programming

by : Lee Adams

* Intranet Resource Kit

by : Dr Prakash Ambegaonkar .

المحتويات

رقم الإيداع : ٩٧٦٦ / ٢٠٠٠

I.S.B.N: 977 - 01 - 673 - 2



هذا هو العام السابع من عمر «مكتبة الأسرة»..
ومنذ سنوات طوال لم يلتف الناس حول مشروع ثقافي كبير كما التفوا حول هذا المشروع الثقافي الضخم حتى أصبح مشروعهم الخاص، وطالبوا باستمراره طوال العام، واستجابة لرغبة المطالب الجماهيري العريض إيمانًا منا بأهمية الكتاب، وبالكلمة الجادة العميقة التي تسهم في إعادة صياغة وتشكيل وجدان الأمة واستعادة دورها الحضاري العظيم عبر العصور.

لقد استطاعت «مكتبة الأسرة».. أن تعيد الروح إلى الكتاب مصدرًا هامًا ورئيسيًا للثقافة في عصر الإبهارات التكنولوجية المعاصرة، وها نحن نحتفل ببدء العام السابع من عمر هذه المكتبة التي أصدرت (١٧٠٠) عنوانًا في أكثر من ٣٠ مليون نسخة تحتضنها الأسرة المصرية في بيوتها وعقولها زادًا وثراءً لأجيال من أجل حياة أفضل لهذه الأمة.. وما زلنا نحلم بكتاب لكل مواطن ومكتبة في كل بيت.

سوزان مبارك

١٥٠ قرشًا

مكتبة الأسرة 2000
مهرجان القراءة للجميع

مهرجان القراءة للجميع
جمعية الرعاية المتكاملة